A Database Publication

ATARI USER

Vol. 1 No. 4 August 1985 £1

The inside story of the revolutionary ST

Write machine code programs with our 6502 assembler

Atari Logo: How to get started with the friendliest language of them all



Complete listings inside

RAIDER 1997 *Futuristic Adventure*

FRUITI GAMBLER *The ultimate fruit machine*

FREE DOS 2.5 UPGRADE
Full details on Page 9



ATARI 1050

COME

RI USER

AZINE.

ATARI USER, STOP HERE.)

A choice of a 1050 Disk Drive or a 1010 Cassette Programmer Recorder and additional software.

No one else could offer you all this power at these prices.

And, as everything comes together, you can make the most of the unbeatable ATARI 800XL straight away.

Without doubt, ATARI Personal Computer Packs are the easiest way to get into computers.

The only difficulty is deciding which one. Now read on.



Pack 2: 800XL;
1010 Cassette Programme Recorder;
Invitation to Programming; Pole Position;
Demonstration Software. £129.99. (Normal cost £194.96).

O NAL COMPUTER PACKS



Vol. 1 No. 4 August 1985

Managing Editor: **Derek Meakin**
Features Editors: **Cliff McKnight**
Mike Bibby
Editorial Team: **Alan McLachlan**
Kevin Edwards
Pete Bibby
Technical Editor: **André Willey**
Production Editor: **Peter Glover**
Layout Design: **Heather Sheldrick**
News editor: **Mike Cowley**
Advertisement Manager: **John Riding**
Advertising Sales: **John Snowden**
Editor in Chief: **Peter Brameld**

Editorial: 061-456 8835
Administration: 061-456 8383
Advertising: 061-456 8500
Subscriptions: 061-480 0173
Telecom Gold: 79:MAG001
Telex: 265871 MONREF G
Quoting Ref. 79:MAG001
Prestel Mailbox: 614568383

Published by:
Database Publications Ltd,
Europa House, 68 Chester Road,
Hazel Grove, Stockport SK7 5NY.

Subscription rates for
12 issues, post free:
£12 – UK
£15 – Eire (Sterling only)
£20 – Rest of world (surface)
£40 – Rest of world (airmail)

"Atari User" welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication by Database Publications Ltd will be on an all-rights basis.

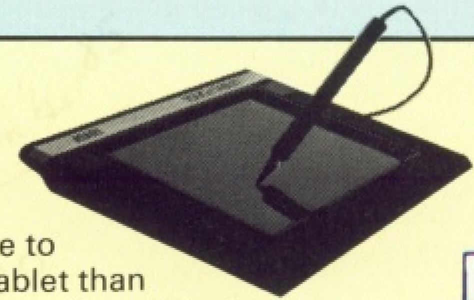
News trade distribution:
Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Tel: 0293 27053.



Contents









ATARI

EUROPES LEADING 'ATARI' MAIL ORDER SPECIALISTS

SOFTWARE EXPRESS

INTERNATIONAL

HARDWARE

130 XE COMPUTER 128K	169.99
800XL/1010 PACK including Software	129.99
800XL/1050 PACK including Software	249.99
1029 DOT MATRIX PRINTER	199.99
WS2000 MODEM	149.55
MODEM INTERFACE c/w Viewterm software	59.95
JOYSTICKS	
PADDLES	

SIMULATION

Sublogic F S. 11	(D)	44.95
F 15 Strike Eagle	(D)	14.95
Solo Flight	(C/D)	9.95/14.95
Space Shuttle	(R)	19.95
Spitfire Ace	(C/D)	9.95/14.95

STRATEGY

Broadsides	(D)	33.95
Colosuss Chess 3.0	(C/D)	9.95 / 12.95
Combat Leader	(C/D)	14.95
Computer Ambush	(D)	49.95
Cosmic Balance	(D)	33.95
Cosmic Balance 11	(D)	33.95
Eagles	(D)	33.95
Field of Fire	(D)	33.95
Fortress	(D)	29.50
Nato Commander	(C/D)	9.95/ 14.95
Odessa Chess 7.0	(D)	53.40
Rails West	(D)	62.75
Reforger 88''	(D)	45.95
Sargon 11 (Chess)	(D)	16.95

FROM THE SOFTWARE FACTORY

MATHS FOR CSE/GCE 'O' Level
A set of Five programs (available individually) which not only test proficiency but also instruct

Years 1 - 4, & 5 (Revision) c/d each	9.95 / 12.95
Centronics Printer Interface	Soon
Character Editor / Animator	Soon

BOOKS

De Re Atari	17.00
Your Atari Computer (inc.XL)	
Mapping the Atari	12.60
Atari Basic Source Book	12.60
Atari XL Handbook	5.95
Technical Reference Notes	17.00
Arcade & Graphics Design	12.80
Atari Basic Faster & Better	16.95
Atari User Encyclopedia	17.20

+ 75p post & packing

ENTERTAINMENT

BC's Quest for Tires	(R)	14.95
Boulder Dash	(C/D)	14.95
Bio Defence	C/D)	9.95 / 14.95
Bruce Lee	(C/D)	14.95
Conan	(D)	14.95
Dig Dug		9.95
Dropzone	(C/D)	9.95 / 14.95
Decathlon	(R)	19.95
Ghostbusters	(D)	14.95
H.E.R.O.	(R)	19.95
Kissin' Kousins	(C)	6.95
Fort App	C/D	9.95 / 14.95
Miner 2049'er	(C)	12.95
Mr Do	(C)	9.95
Pac Man	(C)	9.95
Pitfall 11	(R)	19.95
Quasimodo	(C/D)	9.95/14.95
Smash Hits 1	C/D	14.95 / 17.95
Smash Hits 11	C/D	14.95 / 17.95
Smash Hits 111	C/D	14.95 / 17.95
Snokie	C/D	9/95 / 14.95
Summer Games	(D)	33.70

ADVENTURE

Adventure Quest		9.95
Beyond the Infinite		POA
Cut Throats	(D)	29.95
Deadline	(D)	37.95
Dungeon Adventure	(C)	9.95
Emerald Isle	(C)	6.95
Enchanter	(D)	37.95
Hitch-hikers Guide to the Galaxy	(D)	29.95
The Hulk	(C/D)	9.95 / 17.95
Infidel	(D)	39.95
Lords of Time	(C)	9.95
Mask of the Sun	(D)	43.95
Midwinter	(C)	7.95
Saga 1 - 5	(D)	17.95
Seastalker	(D)	32.95
Sorcerer	(D)	39.95
Starcross	(D)	32.95
Suspect	(D)	37.95
Suspended	(D)	37.95
Wishbringer (Infocom)		JULY
Witness	(D)	37.95
Zork 1, 11, 111		29.95

INFOCOMS INVISICLUES

Hitch Hikers Guide	6.95
Cut Throats	6.95
Suspect	6.95

Available Now
ALL OTHER TITLES ALSO AVAILABLE AT THE SAME PRICE

UTILITY

Basic Routines	C/D)	7.95/ 11.95
Action	(R)	75.00
Action Tool Kit	(D)	27.95
Basic XL	(R)	60.00
Basic XL Tool Kit	(D)	27.95
DOS XL		27.95
MAC/65	(R)	75.00
MAC/65 Tool Kit	(D)	27.95
Sparta DOS	(D)	39.95
Basic XE	(R)	75.00

COMING SOON

520 ST
C/w Disc Drive
B/W Hi-Res Monitor 749.99
Omnitrends Universe & Product Guide
Blue Max 2001
Dambusters
Electrician
Kennedy Approach
New York City
Escape from Fractulas

NEW FROM O.S.S.

The Writers Tool (D & R)
State of the Art Word Processor
(So Good Mike's Collared the First One!)
65.00

MAGAZINES

ATARI USER 1.00 PAGE 6 .90
ANTIC 3.50 ANALOG 3.00

SEND SAE FOR OUR PRICE LIST.
31 STONEYHURST ROAD
ERDINGTON, BIRMINGHAM B24 8HA

HOTLINE
(021) 384 5080

800 XL WINS TITLE

ATARI's 800XL has carried off the prestigious Home Microcomputer of 1985 title in this year's British Microcomputing Awards.

It beat off the challenge from the Amstrad CPC464 and Sinclair's Spectrum+ in the final to get the nod from the judges.

Rob Harding, Atari UK's sales and marketing boss, is seen right receiving the trophy from Matt Nicholson, editor of What Micro?, the magazine which sponsors the award.

The presentation took place at a Hollywood style award ceremony hosted by Sir Alastair Burnett in the Park Lane Hotel, London.

Recognised as the "Oscars" of the computer industry, the event this year attracted more than 1,000 nominations.

Organised by Personal Computer World, the Sunday Times and Thames Television, the awards "seek to define technological excellence and value for money for the consumer".

While pointing out that What Micro? sponsored the award but did not judge it, Matt Nicholson did confide to *Atari User* the reason his magazine nominated the 800XL for the honour in the first place.

"The reason we selected the Atari was mainly because of its £130 price tag", he said. "It was just very good value.

"We feel the 800XL is a good computer, with a decent amount of memory, very good graphics and a good range of software that is no longer so expensive.

"That's why we even rate it above a Spectrum".

But the final word was left to an obviously delighted Rob Harding.

"We believe the machine is an unbeatable combination of performance and value for money – and obviously the award judges fully support our view".



Smile of success: Rob Harding and trophy, with Matt Nicholson

Atari bids to topple BBC

ATARI is poised to launch an all-out bid to capture a major share of the education market.

Effective immediately, it is offering all educational establishments a 22 per cent discount on 8 bit hardware packs and software, and 25 per cent on peripherals.

Spearheading the drive to knock the BBC Micro off its perch in Britain's schools and colleges are two bundled offers containing the 800XL – winner of the Home Microcomputer Award for 1985.

Atari Logo System 1 contains the 64k 800XL, Atari 1010 program data recorder, LCSI Atari Logo cartridge, Introduction to Programming through Turtle Graphics, Atari Logo reference manual, Atari Logo quick reference guide, Atari Basic (built in), Invitation to Programming 1, tutorial software, sound and graphics demonstration software, Pole Position racing simulator, plus all leads and power transformers.

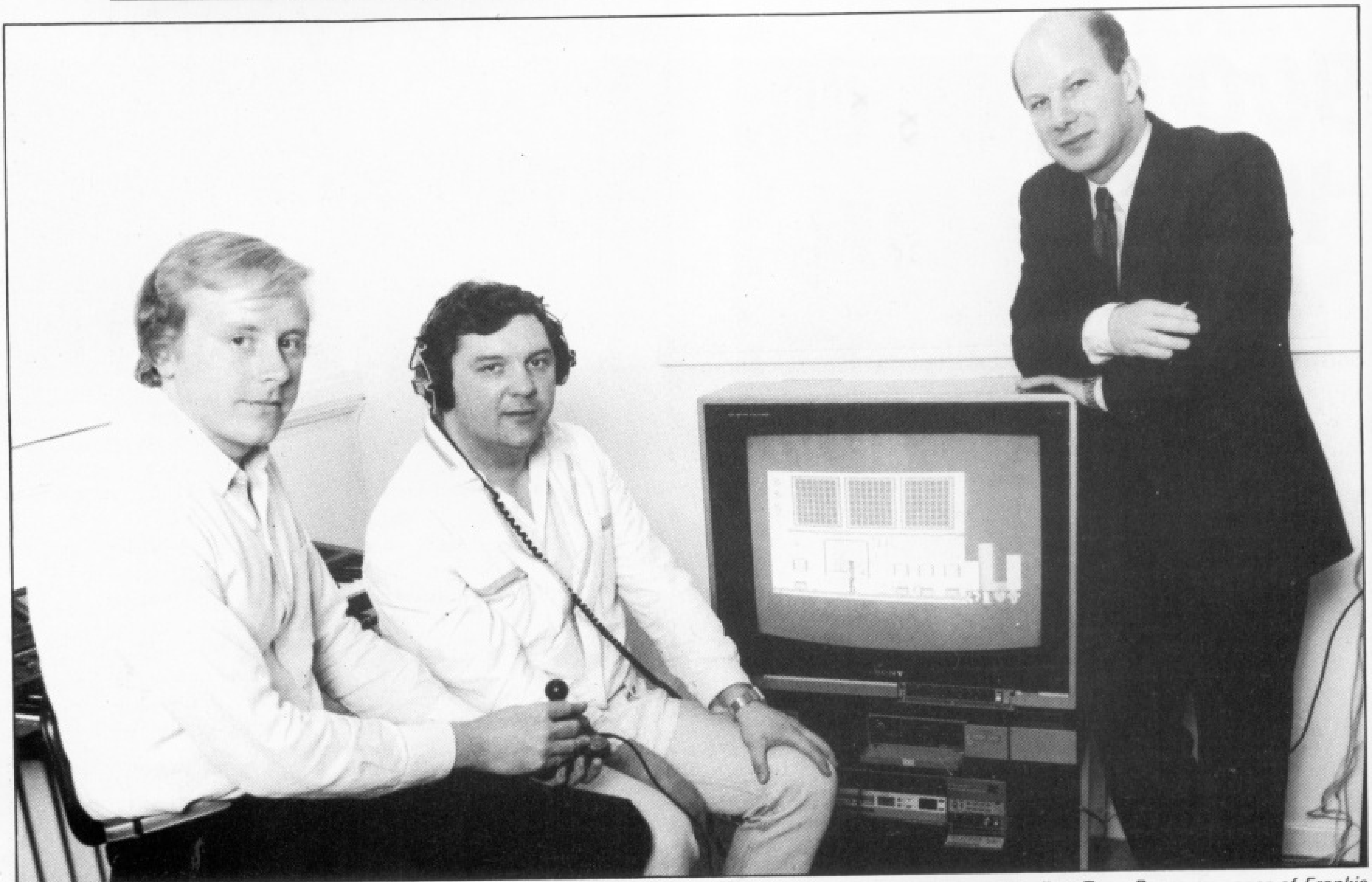
The special educational price is £128.86 compared to the recommended regular retail price of £165.20.

Atari Logo System 2 contains the 800XL, 1050 disc drive, LCSI Atari Logo cartridge, Introduction to Programming through Turtle Graphics, Atari Logo reference manual, Atari Logo quick reference guide, built in Atari Basic, disc operating system software, Home Filing Manager database software, sound and graphics demonstration disc, The Pay Off adventure game, and all leads and power transformers.

Special education price is £210.25 compared to RRP of £269.55.

Atari is also offering educationalists the 130XE for £115.30, the 800XL for £88.17, 1050 disc drive plus DOS software for £130.43, 1010 program recorder for £22.82, 1029 dot matrix printer for £130.43, 1027 letter quality printer for £163.04, Atari graphics touch tablet plus software for £32.60, and Atari LCSI Logo and manual for £40.69.

Atari's Jon Dean said: "We are in the process of finalising our distribution outlets that will serve users of Atari equipment in education. Orders will be processed directly from Atari UK".



Island Records' James Bradley, Tony Pope, manager of Frankie Goes to Hollywood, and Ocean Software director David Ward preview the new game.

IT'S BARGAIN TIME
... with a cheaper model and free software

ATARI says it is developing a cheaper version of the ST – the 260ST. And it could be available here in the autumn.

The operating system with 256k of memory will contain an impressive amount of software on board on ROM, a spokesman told *Atari User*.

No price has yet been set for the machine, which will be released in the UK at the same time that it comes out in the US.

There is a possibility that it could be on display at the PCW Show in September, but this has not been confirmed.

Downloading

FOR the first time ever free telesoftware for downloading to Atari computers has been introduced by Viewfax 258 on Prestel.

Atari owners with Viewterm can copy the software using the built-in downloader in the Miracle Technology package.

The program demonstrates some of the capabilities of Atari's versatile GTIA chip – the television interface chip that converts digital information received from the Antic chip for screen display.

The software is similar to demonstration programs seen in computer shops.

Written by Jerry White, the program uses Basic and machine code and runs on the Atari 800 series and 130XE.

Wait for it

THE fully-integrated spreadsheet/database/word processor package Infinity expected from US developer Matrix Software will not now appear, says Atari.

The package was planned as a much cheaper version of the top-selling 1-2-3 for the ST range.

However, Atari is promising that a "very similar" product will soon be available – "a practically identical package with the same facilities at a similar price, under £100", according to a spokesman.

A treat for pop fans

YOUNG Atari users who have difficulty in making up their minds on whether to spend their pocket money on computer games or pop music have a treat in store.

Ocean Software's latest game program, Frankie Goes to Hollywood, comes with a free audio-cassette containing an unreleased, live recording of Relax.

But – before all the kids go rushing off to the shops – the Atari version will not be available until late summer.

The game has been produced in a joint publishing venture between the group, its recording company Island Records, creative producers ZTT and Ocean.

Says Ocean's David Ward: "Datatune is a new idea – players load the game from the program cassette, and then insert the audio-cassette.

"A voice over will describe how to play the game on side one, and on the flip side players can hear some inspirational music in the form of one of the band's hit recordings".

Frankie Goes to Hollywood is mended selling price of £9.95 for the Atari version, which is the usual price for Ocean's longer-running arcade adventure game programs. It will contain more than 124 screens.

The game's scenario is written around the Frankie philosophy, and the possibility of escape from a mundane existence into the delights of the Pleasuredome.

To gain entrance, the player must grow from a shadowy Frankie figure into a complete 100 per cent person by earning pleasure units.

These are achieved by travelling from a prosaic everyday house, through ordinary living-rooms and kitchens into complex maze situations, and by solving complex puzzles which require both strategy and skill.

Games go on, says Atari

ATARI has denied that it is pulling back from games software production, despite drastically slimming down its programming staff and licensing an increasing number of its titles to independent producers.

The latest game to follow this route is The Pay Off, which was originally produced to promote Atari disc drives. Now its authors, Bignose Software, have gained the rights to produce a cassette version.

But this doesn't mean Atari has lost interest in the games market.

Far from it, says sales boss Rob Harding – "We see a big future for our 8-bit machines like the 800XL and 130XE.

"We are planning further improvements and developments for this range and will be bringing in our own games and small business software.

"In addition we will be encouraging independent software houses to design programs for these machines".

ST programs lining up

AS many as 300 new programs for the Atari ST range could be unveiled at the PCW Show next month.

Development systems have already been delivered to more than 100 UK software houses and most of them are working on more than one program, says Atari.

This means there are almost certain to be at least 200 and possibly as many as 300 ST software items ready for sale or in prototype form at the big autumn show.

Atari expects one-third to be serious business applications, one-third productivity including utilities, and one-third recreational including graphics, design, music and games.

"We are making sure the ST software comes not only in a wide range but covers all aspects of a variety of applications", Atari's Rob Harding said.

Reason why...

AMERICAN program writers' interest in Atari computers is so high that 400 software developers attended Jack Tramiel's recent address to the Software Publishers Association.

The SPA's executive director, Ken Wasch, described Atari's new machines as "the event of the Consumer Electronics Show in Las Vegas".

Tramiel described his activities since he left Commodore and the evolution of his concepts of a new generation of affordable technology.

He let the audience in on the real reason for his going back into the computer business.

"I was in Japan", he said, "and everyone I was talking to was smiling.

"They were thinking that now Jack's out of computers it's time to go into the US".

Wasch said: "Jack Tramiel's enthusiasm was contagious. A broad range of software developers want the machines to succeed.

"If Atari fulfils Jack's promises I think these software publishers would be crazy not to take the bait".

ATARI USER READER OFFER

DOS 2.5 upgrade – and it's free!

"IF you've got a disc drive and currently using DOS 3, then you should think very seriously about switching to 2.5 as soon as you can get your hands on a copy".

THAT'S what our technical editor André Willey wrote in last month's *Atari User*, when he gave an enthusiastic review of Atari's new operating system.

DOS 2.5 offers many advantages over DOS 3, which was issued with the Atari 1050 enhanced density disc drives – particularly ease and convenience of use and compatibility with Atari DOS 2.0. It also includes several utilities, including a Diskfix, a DOS 3 to 2.5 file converter and a Ramdisk for use with the 130XE.

The DOS 2.5 disc also features a "Mini Manual", explaining in detail how to use the new DOS. This can be read or printed using the Atariwriter word processor.

Alternatively, for people without Atariwriter, an additional program has been included which displays the "Mini Manual" from Basic, either on the screen or a suitable printer.

Atari User is happy to be able to offer the new DOS 2.5 to readers in one of three ways:

- ***Send us a blank disc, together with a return postage stamp and the coupon below giving your name and address. Make sure that the disc is adequately packed.*** *There is no charge for this service, but it is limited to one disc per coupon.*
- ***Order our Disk Doubler (details on Page 60) and we will send you, in addition, a brand new disc containing DOS 2.5 completely free of charge.*** *Please use the order form on Page 61.*
- ***Send £1.50 and we will supply you with a new disc containing DOS 2.5. The price includes post, packing and VAT.*** *Please use the order form on Page 61.*

FREE VOUCHER FOR ATARI DOS 2.5

Name ______________________

Address ______________________

____

____

SEND TO: **Atari User, DOS 2.5 Offer, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.**

Please allow 28 days for delivery.

Spielberg epic goes on disc

AMERICAN publisher Datasoft's latest releases include five titles for the Atari XL to be distributed here by US Gold.

Foremost is an action/strategy game based on Steven Spielberg's latest blockbuster film "The Goonies" featuring eight maze-type screens of increasing difficulty with pirate's treasure as the goal.

It will cost £9.95 on cassette and £14.95 for the disc version.

Another all-action game is based on the legendary character Zorro whose adventures take him through 15 screens. Prices £9.95 for cassette and £14.95 for disc.

Datasoft has also acquired the licences for Pole Position II and Elevator Action, two of the most popular arcade games in recent years. Prices £9.95 for cassette and £14.95 for disc.

Alternate Reality is a seven part series of fantasy role-playing games, and The City is the only one that must be bought in order to play the others – The Dungeon, The Arena, The Palace, The Wilderness, Revelation and Destiny.

Game play is controlled by a combination of keyboard and joystick and it will be available on disc only for £19.95.

* * *

SOFTWARE for the Atari ST series will include utilities from Rising Star Industries according to reports from California.

The firm is understood to have signed an agreement with Atari for its Valdocs range of software to be distributed with the ST.

First releases are expected to be Valdraw and Valpaint.

* * *

ATARI is to raise an additional $150 million by the summer of 1986 in order to finance its corporate expansion plan.

The corporation also intends to go public some time this summer.

Electronic censor may clean up the bulletin boards

NAUGHTY words of a type that would make even a sergeant major blush are increasingly confronting Atari users who log onto bulletin boards.

All over the UK, systems operators are being forced to devote more and more of their time to erasing electronic graffiti.

The obscenity problem has been one that to date has baffled the industry. However according to the latest issue of *TeleLink* – a sister publication of *Atari User* – help may be at hand.

It takes a look at a new Naughty Words Editor which is currently being evaluated by MicroLink, the recently-launched nationwide service for micro users.

The man in charge of the project is 39-year-old Tim Clarkson. He explained to *TeleLink* just how the Naughty Words Editor should work.

"You initially create a text file or glossary of naughty words or phrases", he is quoted, "so when these turn up in any message the whole of the text is pulled out and put in an abeyance file.

"Later the messages are checked over by the sysop to see whether it's safe for them to be released".

That's the theory behind it, but in practice – according to *TeleLink* – it has been presenting more than a few problems.

First of all the systems operator needs to have the vocabulary of a drunken sailor.

The second, and possibly the major pitfall is that certain obscenities can crop up quite harmlessly within words.

One of the worries that faces Tim Clarkson is what to do with the species most cherished by birdwatchers, the tit.

"Used in the ornithological context, the word could in no way give offence", he says. "However once it becomes anatomical then eyebrows would understandably be raised.

"So you decide to err on the side of caution and classify 'tit' as a word that might possibly offend.

"What happens then however is that all messages containing with word 'title' suddenly find their way into the abeyance file".

In order to counteract this, the MicroLink arbiter of good taste has created a text file of phrases – and not individual words.

"This removes part of the problem", says Tim. "We can rule that tit is left in as long as it has 'blue' or 'crested' in front of it but not 'big'."

It's victory all round . .

A RECENT US court battle between Atari and Commodore over alleged theft of trade secrets has ended with both sides claiming victory.

Four engineers who had followed Jack Tramiel from Commodore to Atari were acquitted of charges that they took company secrets with them.

The judge said Commodore had failed to prove that the men took any specific plans of the new Z8000 disc drive with them, but ordered the engineers to return to Commodore "certain papers which happened to be in their pockets when they left".

The defendants' lawyer said they were "extremely pleased". The judge had "recognised that these four fellows did not steal any computer secrets and Commodore could not prove they did. This was a complete victory".

Commodore disagreed. Vice-president Joseph Benedetti said: "The judge clearly ruled that they wrongfully appropriated our property. It was a complete victory for Commodore".

32 bit micro on way

ATARI has confirmed it is working on a 32 bit CAD-CAM computer for release later this year – a VAX-type, mini-powered micro whose selling price has yet to be set.

Reports originating in the US said Atari is aiming to put mini power into a micro selling "at around the $5,000 mark, at which it will go like hot cakes".

Now let's get these variables down to work

Expand your knowledge of programming with Part IV of MIKE BIBBY's guide through the micro jungle

WE saw last month how to label strings with variables. This meant that if we were using a string several times in a program we could use a variable instead of it.

For example:

A$=''AUSTRALIA''

means that, from now on, instead of using ''AUSTRALIA'' in full in our programs, we can use *A$*.

PRINT A$

will print out AUSTRALIA for you. Of course we had to make room for the string by telling the Atari its maximum size with a DIM statement.

The labels we used last month were all single letters of the alphabet followed by $. The dollar sign tells the computer that it is a string we are labelling – such a variable is called a string variable.

It is called a variable because the ''contents'' of a variable (in technical terms, its value) can vary throughout a program. Program I should illustrate the point.

```
10 REM PROGRAM I
20 DIM A$(9)
30 A$="AUSTRALIA"
40 PRINT A$
50 A$="AMERICA"
60 PRINT A$
70 A$="AFRICA"
80 PRINT A$
```

Program I

As you will see when you RUN it, the value of *A$* varies as we reassign it during the program. *A$* always takes the last value assigned to it.

You may wonder why on earth you would want to use the same variable for different things, rather than label everything separately. As we shall see, it can be extremely useful.

So far we have restricted our string variables to single letters of the alphabet followed by the $ sign, such as *A$*, *B$* and *C$*.

However there is no need for such a limit – provided we follow them with $. String variables can be made up of several letters, even words. They must, however, be capitals.

Program II illustrates the point. It is our most sophisticated program to date, and is well worth having a close look at.

Incidentally, remember to enter NEW between programs.

Perhaps the first thing to remark upon is that our string variables, instead of being single letters, have grown into actual words. They've still got the $ at the end, though, to show that they're string variables, or labels.

```
10 REM PROGRAM II
20 DIM NAME$(20),FACT$(20),THREAT$(20)
30 PRINT CHR$(125)
40 NAME$="Mr.Smith"
50 FACT$="You owe me money."
60 THREAT$="Pay up or else."
70 PRINT
80 PRINT "Dear ";NAME$
90 PRINT FACT$;THREAT$
100 PRINT "cordially yours,"
110 PRINT "Mike"
```

Program II

Also, notice that while our labels are in capitals, the strings themselves, inside the quotes, are a mixture of lower and upper case. You'll need some deft manipulation of the Caps key as you type it in.

As you'll probably remember, the PRINT CHR$(125) of line 30 clears the screen. It is good programming practice to use words for variables, since we can make the label describe

what it is labelling. Programs make more sense this way.

Thus we use *NAME$* to label "Mr. Smith", *FACT$* to label "You owe me money", and *THREAT$* for "Pay up or else".

This may seem long-winded, but it really does help to make your programs more readable, and hence easier to decipher. For example:

80 PRINT "Dear "NAME$

really tells you what the line is doing, far more than:

80 PRINT "Dear "A$

Similarly:

PRINT THREAT$

is more meaningful than

PRINT B$

The moral is, use words for variables (labels) as much as possible.

Actually, you can use capital letters and numbers intermixed for variable names. For example:

NAME1$
R2D2$
C3P0$

are all valid string variables.

However they must start with a letter – not a digit – and only capital letters are allowed. This means that:

1DAY$
2MORROW$

aren't valid.

Also, spaces aren't allowed, so:

FIRST NAME$

is illegal.

Variables shouldn't start with Basic keywords, as they confuse the Atari, so:

PRINTER$

is definitely out.

Try entering a program line such as:

10 PRINTER$ = "EPSON"

Then LIST it – can you explain what happened? Steer clear of keywords in variable names.

While we're at it, try entering:

10 WRITER$ = EPSON

Spot the deliberate mistake? Well, the Atari does and rejects the line – EPSON should have been in quotes. If you now enter LIST, you'll see the Atari has actually included line 10 as a program line – with ERROR in front of it.

This habit of the Atari can be rather irritating, but don't forget, you can get rid of a line by simply typing its number and pressing Return.

Although it's not likely to affect you at this stage, the Atari limits you to 128 variable names. The good news is that they can each be up to 120 characters long.

One advantage of using variables instead of directly using strings is that we can easily alter the output of the program.

In the case of Program II, if we want another victim to be the recipient of our letter, just change line 40. For example:

40 NAME$="Mr. Jones"

From then on all uses of *NAME$* in the program will refer to Mr. Jones.

In this short program it doesn't make a great deal of difference, but in larger ones, if you had used the string "Mr. Smith" every time, instead of *NAME$,* you would be in for a lot of retyping.

So far we have talked about string variables. However there is another kind of variable called a numeric variable.

Numeric variables are labels just as much as string variables are, only they label numbers in such a fashion that we can do sums with them. Try running Program III.

Line 30 uses the numeric variable *A* to label the number 10. Notice that for a numeric variable we can simply use a letter of the alphabet without following it with the $ sign necessary for a string.

Also since it isn't a string, the value we are giving the variable doesn't have to be in quotes. Hence line 30 is simply:

30 A=10

Line 40 prints out, not *A,* of course, but the value that *A* labels, which is 10.

```
10 REM PROGRAM III
20 PRINT CHR$(125)
30 A=10
40 PRINT A
50 PRINT 2*A
```

Program III

The most interesting part is line 50. Here we multiply the number that *A* labels by two, so that the line prints out 20.

That's the useful thing about numeric variables – you can do sums with them!

Try running Program III with the following versions of line 50:

50 PRINT A+8
50 PRINT A/4
50 PRINT A*A

If you've been following what I've said so far you could be forgiven for thinking that string variables are for labelling words, and numeric variables for numbers.

```
10 REM PROGRAM IV
20 DIM A$(10)
30 PRINT CHR$(125)
40 A$="10"
50 PRINT A$
```

Program IV

Life is never that simple. You can, and often do, use string variables for labelling numbers – the point is that you can't do sums with them. Try entering Program IV, which is based on Program III, using the string *A$* instead of the numeric *A*.

Once you've entered it, try adding the following line:

50 PRINT 2 * A$

As you'll soon find out, the Atari rejects line 50 out of hand. This is because you are attempting to do a sum with the wrong type of variable – string instead of numeric.

As with string variables, we do not have to (and should not) restrict ourselves to single-letter labels for numeric variables.

We can use words in a manner strictly analogous to string variables, save that we omit the final $ sign. And, of course, we don't put what we are labelling in quotes, since it isn't a string.

Have a look at Program V. This is meant to be a cheery greeting for

```
10 REM PROGRAM V
20 PRINT CHR$(125)
30 DIM NAME$(10)
40 NAME$="MIKE"
50 PRINT "GOOD TO SEE YOU, ";NAME$
```

Program V

someone when they RUN the program in the computer – the sort of thing I often used in my classes.

However as it stands it's a bit restricted – after all, only a small percentage of my students were called MIKE. What's really needed is some way for the Atari to find out the name of the person so that it can tailor the message to suit.

Program VI fits the bill. The trick here is the use of the INPUT statement in line 50. In Program V, line 40 put the value MIKE into *NAME$*. In Program VI the variable isn't actually attached to a specific

```
10 REM PROGRAM VI
20 PRINT CHR$(125)
30 DIM NAME$(10)
40 PRINT "WHAT IS YOUR NAME";
50 INPUT NAME$
60 PRINT
70 PRINT "GOOD TO SEE YOU ";NAME$
```

Program VI

value – if you like, you give the program a label, but neglect to tell it what it's labelling. Instead you type:

50 INPUT NAME$

When the Atari reaches this line it waits until you PUT IN, or INPUT, the value you want *NAME$* to have by typing the value in.

To put it another way, when the computer meets an INPUT statement followed by a variable, it asks you what you want the variable to be – in fact, it actually puts a question mark on the screen.

You are then supposed to type in the answer followed by Return, which, as always, sends it to the computer, which then carries on with the rest of the program.

So when you run Program VI line 40 asks: "WHAT IS YOUR NAME". Notice that we don't need a question mark – the INPUT statement of line 50 supplies that.

The micro then waits for us to type our reply and send it by pressing Return. Whatever we have typed in then becomes the value of *NAME$* – even if we have lied!

Line 70 then prints out the message after line 60 prints out a blank line.

The point of all this is that in Program VI, as opposed to Program V, the value of *NAME$* is not fixed initially, but is decided during the program by the response to INPUT.

This means that every student in the class can now run the program

```
10 REM PROGRAM VII
20 PRINT CHR$(125)
30 PRINT "How old are you";
40 INPUT AGE
50 PRINT
60 PRINT "I don't believe you are ";AG
E
```

Program VII

and have the message tailored to themselves.

Incidentally, line 40 is not strictly necessary, but it is only polite to tell people what kind of response you expect them to make. Otherwise they will be met with just a question mark – not too "user-friendly" as the jargon has it.

The semi-colon at the end of line 40 "glues" the question mark, or prompt, as it is known, to the preceding "message". Running the program with it omitted should make this clear.

Remember, when you run Program VI and it asks for your name, you must type your reply then press Return. If you omit Return the Atari won't receive your answer and will continue waiting. This could be incredibly boring!

If you make a typing mistake before you press Return, you can erase it with Delete. Once you've pressed Return, though, you're stuck with what you've typed.

You can use INPUT with numeric variables as well as strings. Program VII demonstrates this. When you get the prompt, try typing in a word rather than a number and see what happens.

```
10 REM PROGRAM VIII
20 PRINT CHR$(125)
30 PRINT "First Number";
40 INPUT FIRST
50 PRINT "Second Number";
60 INPUT SECOND
70 PRINT FIRST;" multiplied by ";SECON
D;" is ";FIRST*SECOND
```

Program VIII

A slightly more serious application of INPUT allows you to calculate the product of two numbers, as Program VIII demonstrates.

Look carefully at line 70 and see if you can work out what's happening. *FIRST* isn't in quotes, and so the micro will print the number that *FIRST* labels. "Multiplied by" is printed literally since it is in quotes.

The numeric variable *SECOND* is not in quotes – it may have them on either side, but the quotes on the left are already paired with the quotes on the far left, so they don't count. The micro will therefore print out the value of *SECOND*.

"is" is printed literally, since it is in quotes. *FIRST*SECOND* isn't in quotes, so the sum is done and the answer printed out. Figure I should help to make this clearer.

Finally, try altering Program VIII so that it adds or subtracts pairs of numbers.

We've covered an enormous amount of ground this month. I suggest that you spend a good while going over the programs. If you are having problems, re-reading the earlier articles will probably help.

Above all, remember it's a "hands-on" course – you can't expect the examples to make sense until you've typed them in!

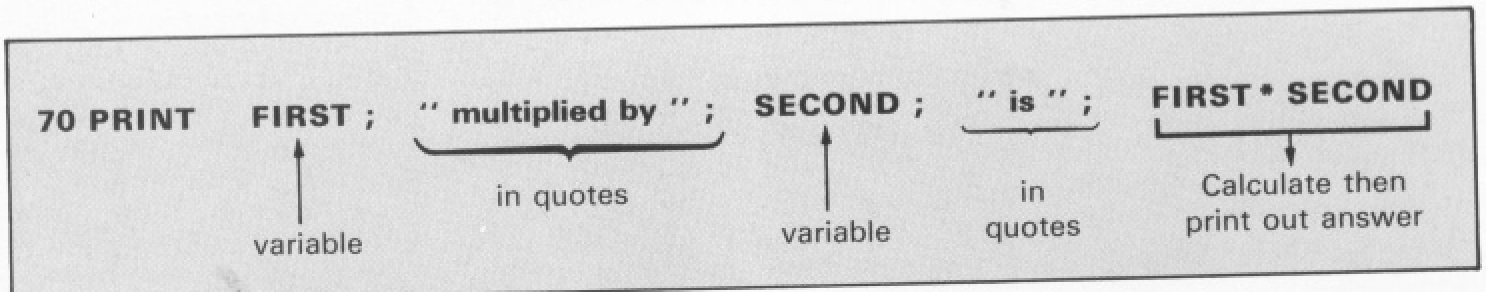
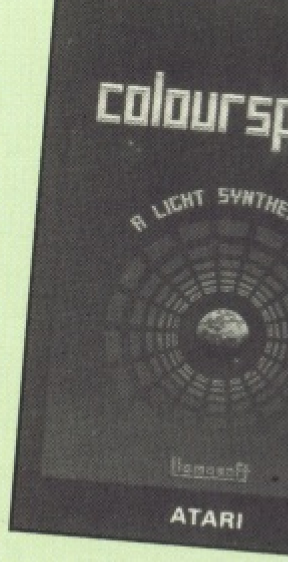


Figure I: Mixing variables and strings in PRINT statements

Psychedelia by any other name is just as good



THE first couple of issues of *Atari User* carried an advert for Llamasoft's Psychedelia.

If you're one of the many people who've been eagerly waiting for Psychedelia to appear on the Atari I have some bad news and some good news.

The bad news is that Psychedelia isn't going to appear on the Atari. The good news is that Jeff Minter was so pleased with the Atari version he's called it something different – **Colourspace.**

Let's get one thing clear from the start. Colourspace is not a game, unlike the rest of Llamasoft's catalogue. It's what Jeff calls a light synthesiser, a software toy.

It's also nigh-on impossible to describe.

If you can imagine an etch-a-sketch connected to a multi-coloured special effects generator, you might be on the right lines.

You "play" it with a combination of the joystick and keyboard to produce incredible coloured displays.

They can be whirlwind rainbows or cool waves flowing endlessly – it's up to you.

As Jeff points out in a glowing eulogy to Atari machines, all this is possible because the display list can tell the Antic chip to build any number of screens.

According to the manual, the difference between Psychedelia and Colourspace is as pronounced as the difference between a Mini and a Ferrari.

The idea for Colourspace grew out of Jeff's interest in rock music and the light shows that accompany rock concerts of the Pink Floyd/Genesis type.

A very comprehensive manual accompanies the tape. It's written in the distinctive Minter style – "imagine that the cursor is a telepathic metagoat" – but actually does describe all the many controls with which the parameters are set or manipulated.

I must admit I didn't read the manual at first. I glanced through while the program was loading and then played for quite a while.

I then kept dipping into the manual to discover a new variable and immediately try it out.

The crowd that gathered around sounded like kids on bonfire night. Lots of "Oohh"s and "Aahh"s, with a liberal sprinkling of "How does he do that?"

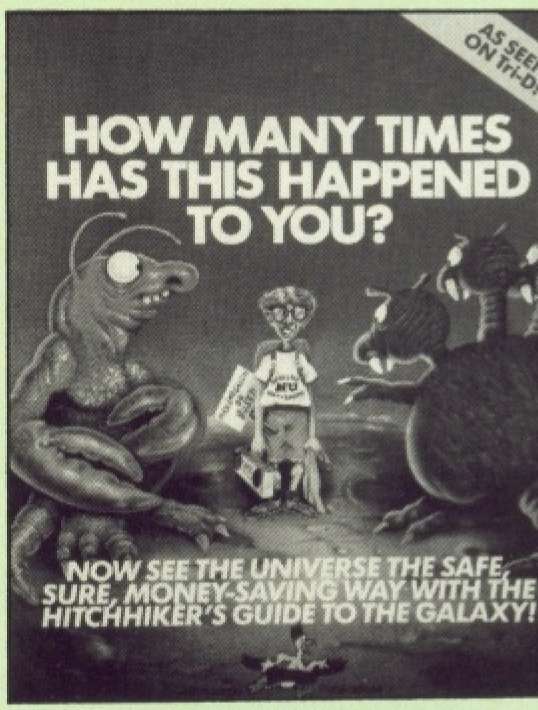
My daughters enjoyed it even more. It has all the compulsiveness of a kaleidoscope with all the fun of being able to tweak the controls.

You can even record a sequence of about 15 minute's worth in memory and play it back as an endless loop, or record the parameters and joystick/keyboard dynamics to tape and load them back in at a later date.

Quite simply, Colourspace is magnificent. It's Atari graphics at its best and no hippy should be without it. Nice one, Jeff – really zarjaz.

Cliff McKnight

HITCH YOUR ATARI TO A STAR GAME



WHAT sort of probability factor would you give to the chances of a cult radio programme going on to become a television series, an LP record, several books, a stage show and is currently being made into a movie?

Highly improbable, right?

Well **The Hitchhiker's Guide to the Galaxy,** by Douglas Adams, has not only achieved all of that already, it has gone one step further.

The immensely successful series now features in a brilliant text adventure, written by Douglas Adams himself and programmed by those masters of artificial intelligence at Infocom.

And believe me the result is magnificent. It has already gone straight to the top of the charts and has just picked up the W.H. Smith Game of the Year award, probably the first of many such awards.

Hitchhiker looks set to be one of the all time greats.

Like all Infocom adventures, it is text only, has an immense vocabulary, an amazingly sophisticated input analyser, screens and screens of fulsome prose, and, because of the sheer size of the game, comes on disc only.

Even if you've tasted the sweet pleasures of an Infocom adventure before, I guarantee

→

you'll never have played one like this.

When was the last time you suddenly found yourself transformed into another character partway through the game and found yourself talking to yourself, if you catch my drift?

And that doesn't happen just once, either.

You begin the game as Arthur Dent. Your immediate concern is how to stop the local council bulldozing down your house in order to make way for a by-pass.

However that anxiety soon becomes a trifle insignificant since the Earth itself is about to be destroyed by a Vogon Constructor fleet to make way for a galactic by-pass.

If you are familiar with the books, or radio series, etc, you'll find the opening sequences ringing a few bells. But you can't rely on that knowledge for very long – you are soon confronted with many situations that are going to take more than a little lateral thinking to resolve.

Many of the characters from the series make an appearance. Ford Prefect, Zaphod Beeblebrox, Trillian and Eddie, the ever-cheerful shipboard computer. And, of course, the galaxy just wouldn't be complete without Marvin the paranoid android.

He's still as miserable as ever and his behaviour will surely make you a little paranoid, too.

There's also a host of much-loved subsidiary characters, objects and incidents. Remember the Ravenous Bugblatter Beast of Traal? He's still ravenous and dangerous but very stupid – if you can't see him, he thinks he can't see you.

The awful Vogon captain with his even more awful poetry is here, and so is the Babel fish, the obtaining of which, incidentally, presents one of the most devious but deliciously amusing, multi-layered puzzles I have ever encountered.

It's almost as if the game is outthinking your every move.

I am not at liberty to reveal just what your ultimate goal in the game is, not that it would help you in the slightest if I did.

But there is one source of help available throughout the game and that's the guide itself.

By typing CONSULT GUIDE ABOUT something, chances are you will glean some useful, and certainly hilarious, information which may, or may not, assist you in your mission.

And even when the guide cannot provide data on the selected topic, you're still sure of a variety of witty responses.

If you really get stuck in the game, don't panic. You could do a lot worse than invest a further £7 in a copy of Infocom's Invisiclue book – concealed hints – for the game.

It is cunningly designed, entertainingly written and great fun in itself. It not only offers help where needed but provides lots of other suggestions to try out when you've finished the game, many of which might never have occurred to you.

The book really does help you to get every last ounce of enjoyment out of the adventure. Only buy it when desperate for help or when you've completed the game, as the temptation to consult the clues is overwhelming.

The game comes with a comprehensive manual and includes your very own piece of fluff, pair of peril-sensitive glasses – totally black – and a microscopic space fleet. You must supply your own towel.

Hitchhiker is zany, original, challenging and entirely and faithfully logical in its own crazy world of logic.

The chances of you finding as funny or as superb a game as this between Earth and Magrathea are two to the power of ten million and rising, so don't bother waiting – hitchhike to your nearest dealer now.

Bob Chappell



Mr Robot sets a tricky scene

I MUST admit that when I first booted **Mr Robot** my immediate thought was "Ho hum, another levels, ladders and power pills game".

Level 1 presented no difficulty other than determining which jumps were permissible and which were fatal. Level 2 wasn't much harder.

Slowly, though, things started to get a bit trickier. The first sign of trouble came on Level 4 with the bombs. They are not active until you walk on them, whereupon they fizz for a few seconds and then explode.

The problem is that you need to walk on them in order to collect all the power pills. It's a one-way journey – once they've exploded there's nothing to walk on. This means that you've got to plan your route around the screen.

Then come the trampolines to bounce you on your way, the transporter tokens which may jump you out of the microwave oven into the central heating boiler, and of course through all this the Alienfire is still intent on your destruction.

Although the game starts off easy, you can select which level you start from. This means that you don't need to work your way up through levels which you've already mastered.

According to the manual only the first five levels are selectable. However in practice you can select any of 22.

Level 13 is a bonus round with no obstacles, and I found Level 14 remarkably easy, but there are plenty of "killer" levels to keep you busy.

If you've mastered the levels and think you could design better, you can give it a try.

In addition to the game there is a DIY section called The Robot Factory that lets you create up to 26 of your own screens. These can be saved to a separate disc.

New screens are drawn by picking up pieces with the cursor and simply putting them where you want them. It's very easy. You can play-test your screen and keep editing it until satisfied.

Having the same elements to build with, my screen came out looking very much like the real game screens. However I must admit I'm not very creative when it comes to such matters.

If you've a flair for design you could probably combine the elements in a more creative way.

Mr Robot is an American import from Datamost and is being distributed in this country by Zoomsoft. It is only available on disc at £14.95.

This may seem a bit expensive. However, when you consider the game's 22 levels and the facilities to build an extra 26 levels it's likely to last you for a bit longer than the average levels and ladders game.

Dave Russell

Carol drops a bombshell or two into a man's world

IN today's male chauvinist world it makes a change to see a game written by a woman – or should I say a female person?

Activision's **River Raid** was written by Carol Shaw and is proof of the fact that women have as much to contribute to computing as men.

The river of the title is divided into sections with a bridge at each end. Your job is to fly your plane up the river and destroy the bridges.

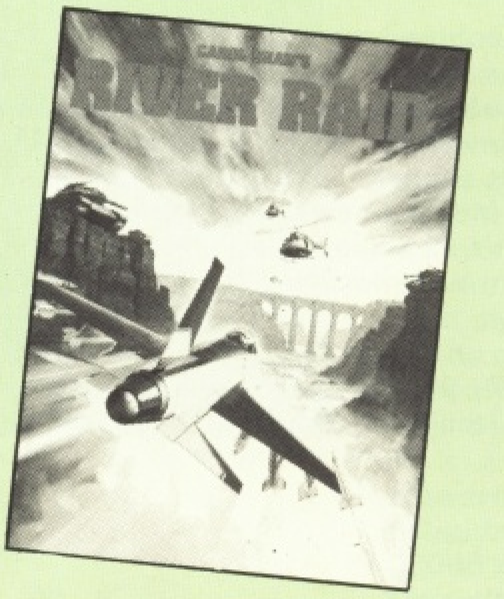
You're constantly moving forward, or rather the screen is constantly scrolling downward, and there are various obstacles to your progress – ships, helicopters and so forth.

You're also using up fuel, but fortunately the river is littered with fuel dumps. You need only fly over them to refuel and you can blow them up and earn points if you don't need to top up.

As you get past more and more bridges the obstacles get more frequent, the nasties get more aggressive and the fuel dumps scarcer.

In fact, as you'll gather from my description, there's nothing that you'd call innovative about River Raid. However, it does have several things to recommend it.

Firstly the game is well implemented. Scrolling is smooth even at high speed, response to the joystick is good without being over-sensitive, and the colours are crisp and clear.



Secondly there are options to start at bridges 1, 5, 20 or 50. This means that once you've got the hang of the game you can leap straight in without having to go through the easy sections.

It's also a very fair game. If you destroy a bridge but get killed off before you fly past it you start your next life from that bridge rather than the previous one.

As an arcade fan I enjoyed River Raid. It's accessible enough at lower levels to allow you to get accustomed to it, but challenging enough at higher levels to hold your interest.

Pat Cookson

Pining for Nevada with Pac-Man

I FIRST discovered Pac-Man in Las Vegas *(What a name-dropper, Ed.)*. While all around me were pumping money into one-armed bandits, I was supporting the local arcade machine leasing company.

It's an addiction that has stayed with me throughout the years. I've played the game on a variety of machines and for more hours than I care to admit and I still love it.

Imagine my delight then, to get a review copy of US Gold's release of **Pac-Man** under licence from Datasoft.

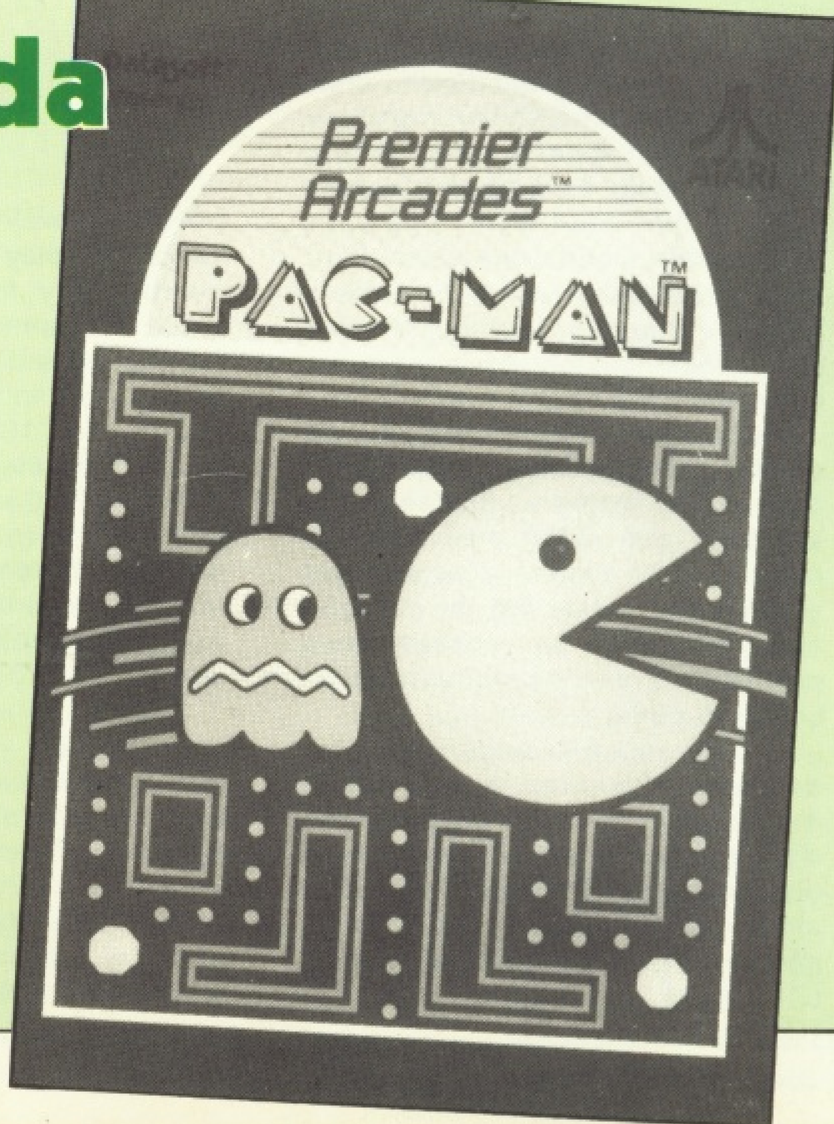
It's described as "the official version of the arcade classic" and is about as close as you can come to the original without actually spraying light ale around the room for effect. I was pining for the Nevada desert after a few games.

If you don't know what Pac-Man is, I hope you've been very happy in the monastery or convent for the last five years. Like Space Invaders, the game has become part of the micro industry folklore, so I shouldn't need to describe it.

Suffice to say that in the unlikely event of your software collection not containing a version of the game, you should buy this one.

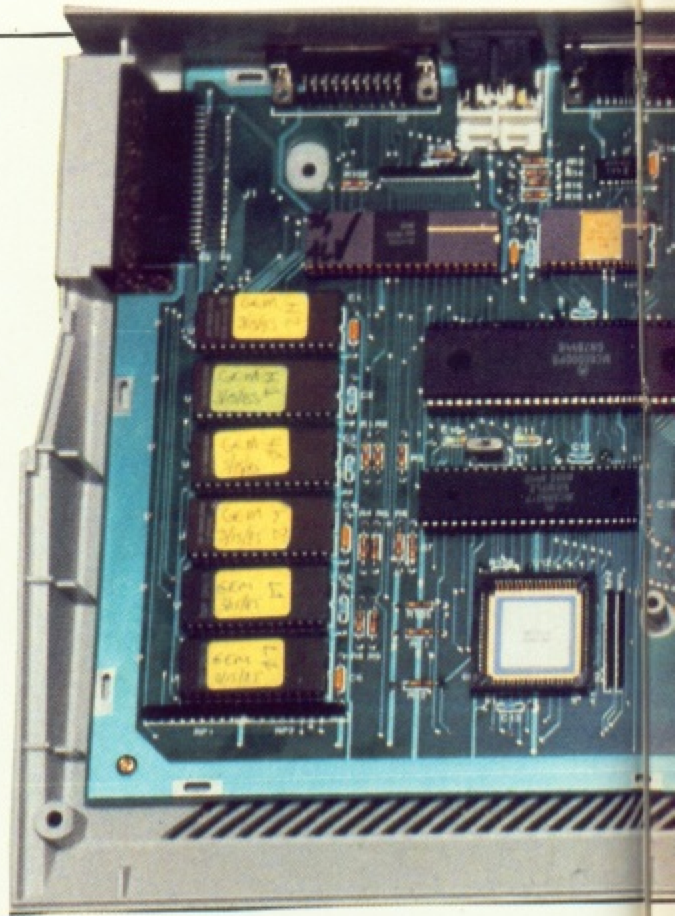
The tape will cost you £9.95 and the disc is £14.95. Either way it's a lot cheaper than going to Las Vegas.

Cliff McKnight



ANDRE WILLEY

takes a long hard look at Atari's new half megabyte superstar model 520ST . . . and likes very much what he sees



THE new range of Atari machines are probably the most talked-about forthcoming items in the home computer world. The American magazines have been bubbling over with enthusiasm, and the expectations built up so far will be hard to match up to. After all, there MUST be a catch, mustn't there?

A 512k machine with the M68000 running at 8mHz, a half megabyte 3.5 inch drive, mouse, GEM with 512 colours, Basic and Logo built in, high-res monitor, Gem-Paint and Gem-Write included . . . and all for £750?

Well, I'm happy to report that it meets and far surpasses all of my expectations Let's first re-cap on the general information about the 520ST. It will be part of a whole range of computers – the new generation of Atari micros – and it seems as far ahead of its time now as the 400 and 800 were when they were released.

SUCCESS, with a capital

The old range used the now long-in-the-tooth 8 bit 6502 chip, but the ST runs on the Motorola 68000 processor. It also runs at a little over four times the speed of the 6502, and has so many functions built in that I go a little green with envy every time I read the chip manual.

Atari has thankfully used the full version of the 68000 – with 16 bit address lines. The Sinclair QL, on the other hand, uses an 8 bit version of the chip – guaranteed to slow programs down dramatically. Perhaps QL stands for Queer Logic?

The main board is superbly designed, as we have come to expect from the new Atari team.

The chip count has been kept down by packing many operations normally requiring a number of chips on to single, custom designed super chips.

I won't cover the technical details of all the ones used, but they include high-speed memory management, graphics and DMA management chips – Atari custom design – two serial output chips for the RS-232 and Midi ports, a separate micro processor to run the keyboard and the on-board clock.

Then there's a Yamaha sound chip complete with three voices, ADSR, a controller for running up to two disc drives, and another for the hard disc interface. Plus six 32k ROM chips containing Gem, Basic, Logo, and so on, and 16 chips to provide the 512k RAM.

The system ROM chips are not yet complete – they should be ready this month, ready for the main release at the PCW show in September – so the machine I got my hands on booted Gem in from disc. Assuming that the full 192k was booted, the floppy disc drive seems quite fast.

Normally Gem will greet you at power-up with its main Desktop window. The concept of "windows" may be new to most of you, but they're basically very simple and useful.

Imagine a window as being a screen display – just like the one you see on your Atari now. However you can have many windows on one display, and move and change them at will.

To do this the mouse comes into play. The Atari mouse is a two-button affair which will copy any movement you make with it on to the screen.

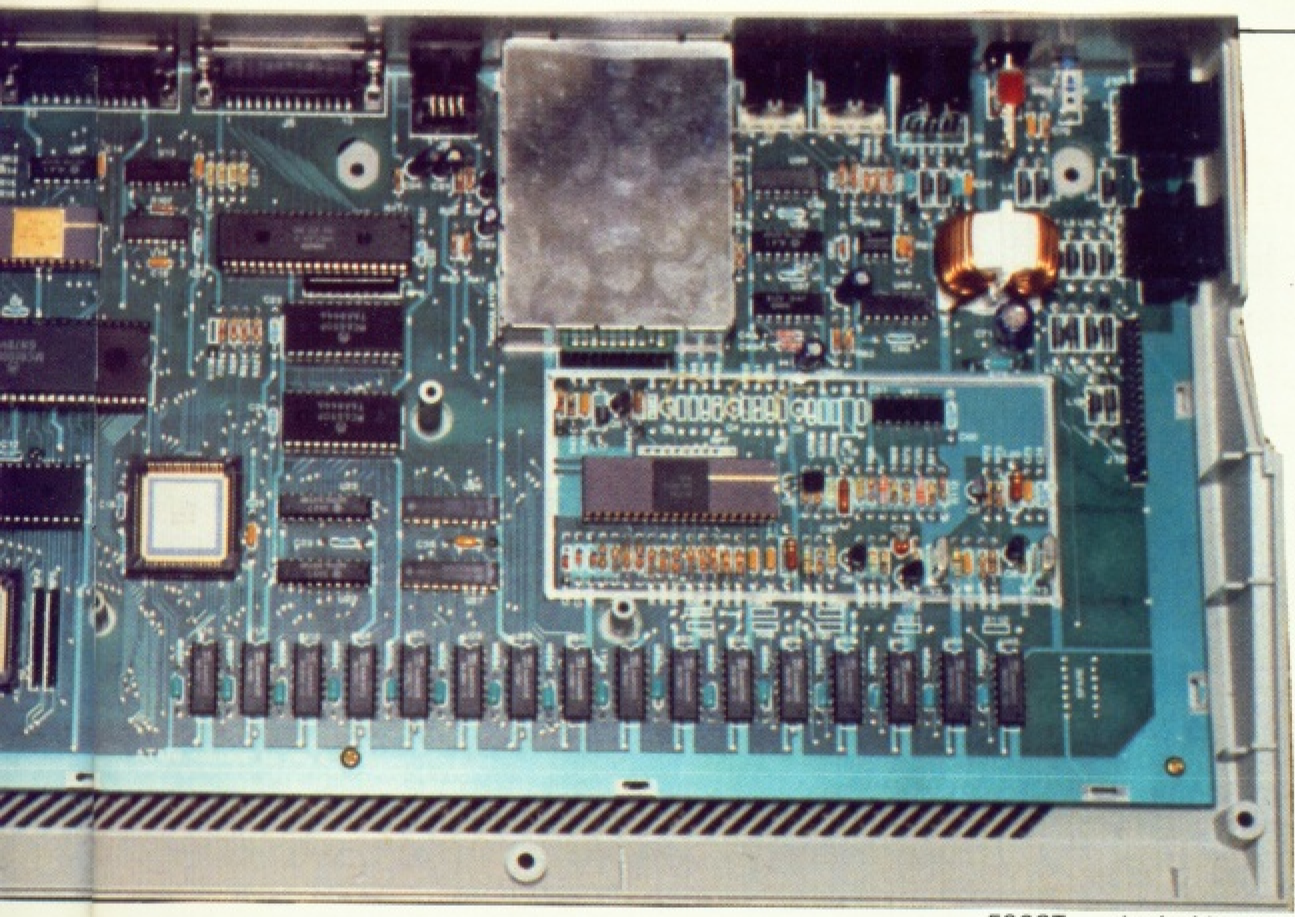
To access a function you simply point the mouse at the relevant icon – computer jargon for picture – and press one of the buttons. Up pops a new window, containing all of your choices for that function.

I only touched the keyboard once during my session with the machine – and that was to test the feel of it. It is a similar type to that used on the 130XE, which I am quite keen on.

In addition to the main qwerty segment, there is a cursor key section and a full numeric keypad.

Incidentally, if your mouse breaks down you can use the cursor keys to control Gem, but normally the mouse makes the system so user-friendly that the claim that anyone could start to use it immediately is not unjustified.

The great thing is that, unlike most



520ST . . . the inside story

ST

user-friendly systems, Gem will not also hold back an experienced user.

Gem itself handles everything that DOS does on your old machine – and a lot more, too. You can get a directory in pictures, or text – and even sorted by name, type, size, or date created. This may be from any attached disc drive, and will be displayed in a window.

If the window is not big enough, or it obscures something else you wanted to read, you can grab the corner with your mouse and drag it all over the screen, change its size – to full screen, if necessary – and scroll the information within the window in any direction.

You can even open another window over the top of it and get the first one back intact whenever you want.

I was, however, most disappointed that Gem does not make toast, and the kitchen sink implementation was rather poor . . .

The icons try to show you what each file is – and you can define your own icons once you get into programming.

A file is displayed as just that – a file.

You can put any number of files into a "folder", and even put folders and files into other folders. This is akin to sub-directories and path-following, but without the hassle this usually involves.

If you want to copy a file from one folder to another you can simply open directory windows for each folder, grab the file you want by pressing your mouse button while over it, drag it into the other window, and release the button.

To copy a file onto another disc, just grab the file and pop it into the icon for drive B. Simple as that. If you decide you don't want a file, just drag it over to the trash can and drop it in. After a quick double-check, bingo it's gone.

Anyone who has had the misfortune to use an IBM to do some of these sort of tasks will realise just how amazing Gem is.

All of these functions could run on a standard TV set, if required, but the ST is capable of much more, and indeed is provided with a high-res monitor as standard.

The lowest display resolution is 320 × 200, which is the same as Graphics 8 with no text window on the current Atari range, but can display up to 16 colours.

Medium resolution, which wouldn't look too good on a TV set as it uses an 80 column display, gives 620 × 200 with four colours.

If you use the monochrome high-res monitor provided with the system you can use the maximum resolution of 620 × 400. This is slightly higher than that of the Macintosh, but only gives you two colours.

The machine will sense which type of monitor you have and adjust itself accordingly at power-up, although you can pull down a menu to change resolution at will and even save your configuration to disc for next time.

The 512 colour palette can be selected from by using another pull down menu, and you just push the Red, Green and Blue sliders to the level you require. No more trying to remember complex SETCOLOR numbers.

Other pull-down functions include a mini-terminal emulator for the RS-232 port, a calculator and notepad, disc set-up menus, and other system configuration details such as clock setting, mouse speed,

→

and even an alarm clock.

I have not had a chance to see Personal Basic on the ST yet, but if it's anything like Logo it should be great. Both Logo and Basic will be in ROM and still support all of the user-friendly features of Gem.

Logo, and Basic, I'm told, has three main windows, and will let you define others as you need them.

You will program in the editor window, see your graphics in the graphics window (surprise, surprise...) and use the dialog window to receive communications from Logo.

As before, Gem will allow you to move and change the windows as required.

Set up and run a program in one window, and while it's running and rushing its turtle all over the place good old Gem can multitask and you can pull the graphics window over the whole screen – all of this with no noticeable difference in speed of program execution.

From what I've neard, DR's Personal Basic will allow the same sort of facilities. Just imagine, programs and graphics wherever you want them – watch the listing while the program is running. It's like having two TVs going at the same time, each having the resolution of full-screen Graphics 8. When final versions of Basic and Logo are ready, we'll give them a full bench test.

Gem is packed with useful facilities for the programmer, far too many to list here, but they include routines for mouse control, window management and disc control.

Another useful facility is the Gem VDI – that's Virtual Device Interface – a way to generalise control of lots of different types of peripheral, so your program can send information to any of them in the same basic format, and Gem will deal with how each unit handles graphics, text, and so on.

This VDI includes routines for handling different text fonts and sizes, graphics of any description – including bar and pie charts, even in 3D, drawn automatically, circles, arcs, ellipses. lines, polygons, pattern-filled areas and much more.

Because VDI is device independent, the same set of commands used to generate the display on screen could be sent to a printer or plotter, in colour if your peripheral handles it.

And if you don't want to bother with that, Gem has a screen dump facility anyway. Configurable to any printer type, of course.

All of these facilities are easily accessible from assembly or high level languages, although I don't know how Basic will interact with Gem. It may have commands for some of the above, but probably many of them will have to be accessed by some form of CALL or USR command.

In addition to these Gem-based features, the machine will support both vertical-blank and horizontal-blank interrupts – useful for synchronising programs to the screen display, and for critical timing requirements.

Sprites are supported by means of the high speed memory management chips, rather than by separate hardware devices.

Screens can also be defined in more than one logical plane for various effects and colour combinations.

Having dealt with how the machine interacts with the user, how does it fare in terms of other contacts with the real world?

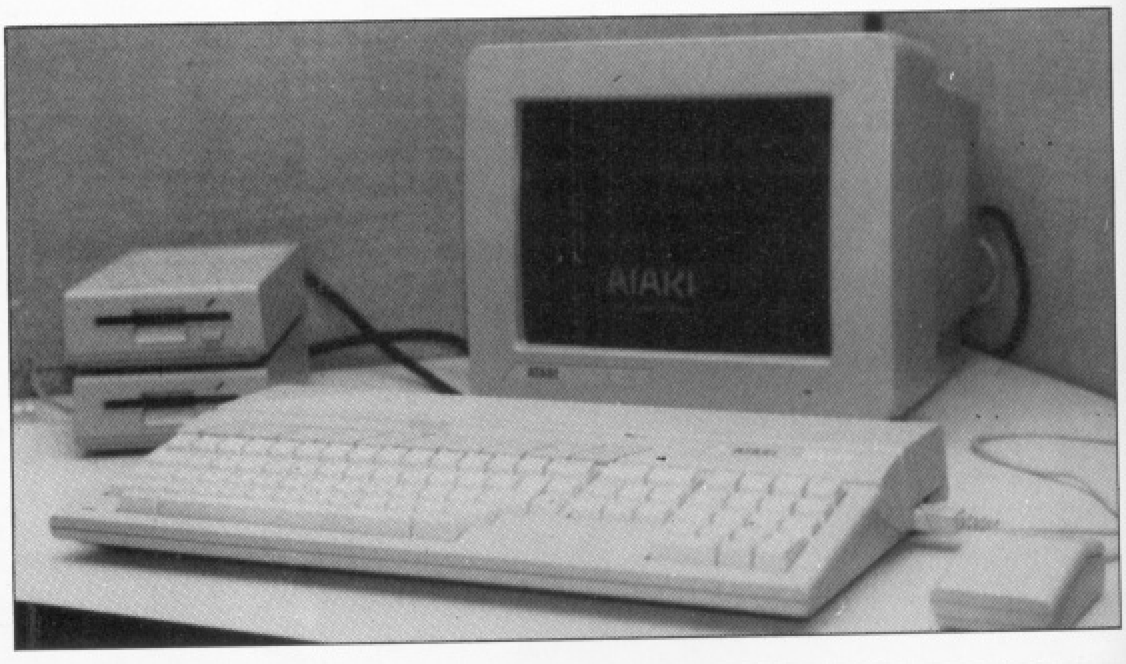
Well the back of the ST is crammed with almost every imaginable connection you could want, bar one.

From left to right, we have the power socket, with adjacent power switch and reset button, two Midi ports – IN and OUT – for computer control of single or multiple synthesisers, the TV output and the monitor output, giving analog RGB, composite colour, high-res monochrome and audio, the printer port, a bidirectional Centronics connector, the RS-232 serial port, the floppy disc port for up to two parallel drives, and finally the hard disc interface – which can supposedly transfer data at an astounding 1.3mbytes per second.

THE ATARI ST RANGE

520ST (512k) *including high-res monitor, half-Meg drive, mouse and bundled software: £750.*
260ST (256k) *with built in 500k drive, mouse and software: £500.*
FS354 (500k) *3.5 inch disc drive: £150*
FS314 (1M) *3.5 inch disc drive: £?.*
Hard disc *10m-20m storage: £400-£500.*
CD ROM *compact disc giving a massive 550mbytes of read-only storage space: £?.*

Software should range from about £50 to something over £100, although I'm sure that there will be lots of games software at very much lower prices.





On the left side of the machine is a ROM cartridge socket capable of taking an extra 128k of ROM. The right side features two joystick ports, one doubling as the mouse port.

That's a lot of connections for any machine, and it's quite astounding to have that sort of versatility on a low price micro. The missing socket I mentioned would be an expansion port for extra RAM.

I know that 512k seems a lot, but there's a rule within the computer industry which states that when writing any given program it will quickly expand to fill all available memory. This applies especially to databases and word processors.

Hopefully some clever company will design a RAM pack to fit either the hard disc port or the ROM socket.

Unfortunately at this stage it is not possible to fully review and test any of the applications software or languages. The development systems being shipped to software houses by Atari do not include Gem-Write or Gem-Paint or even Basic, but these will be available in the next few months.

Software houses, of course, do get such things as a C compiler, linker, 68000 assembler and an editor, plus a few thousand pages of documentation.

It will take them quite a while just to wade through the paperwork, but at least there should be some software under development by now.

Hopefully, according to Atari, anyway, there will be about 100 titles available by September for the PCW Show.

That may be a little optimistic, but assuming the software houses are as enthusiastic as they all seem to be, there should be a few completed programs, mostly business orientated, plus many others in various stages of development.

Now for the 64 million dollar question – when can you buy one? The current position, and this may well have changed *again* before you even read this, is that the first batch of machines, most of which have now been dispatched, went to software houses at about £1,200 for a development system.

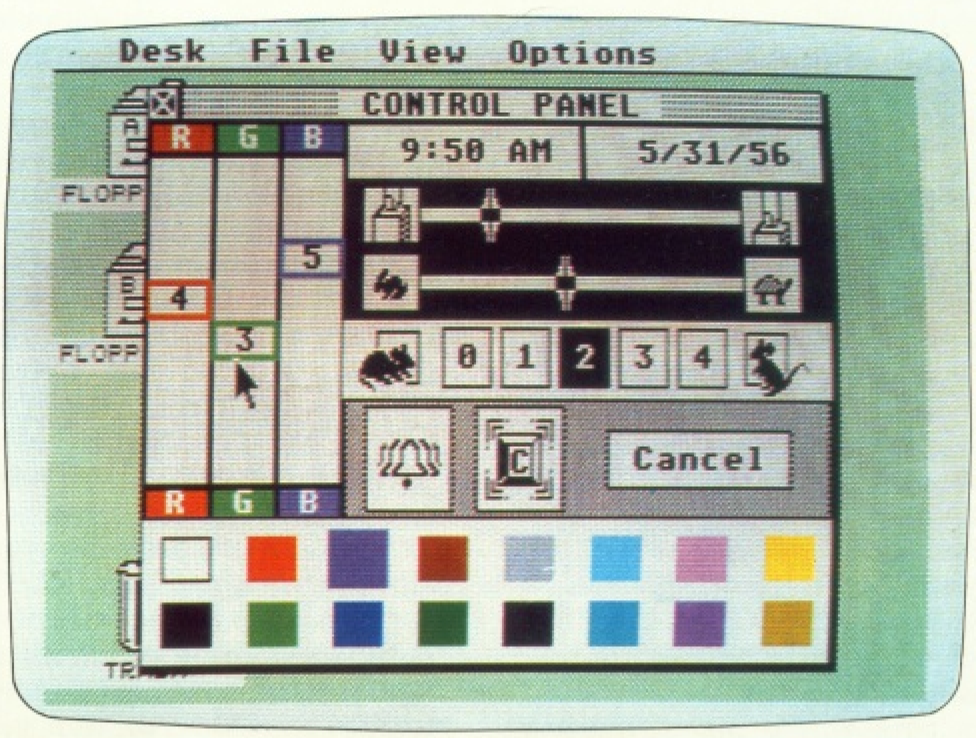
More machines were due to arrive in the UK at the end of July, these being available to major retailers and specialists, but in very limited quantities.

You probably won't be able to actually BUY one of these, but perhaps your local user group can get one, and you can at least have a good look for yourself.

The main launch, by which time a good stock of machines should have arrived, will be at the PCW Show in September, and retailers should get their stock at that time. But who knows? Read next month's thrilling instalment for a complete change of plans . . .

I know that many of you will be waiting, like me, with bated breath to buy an ST, and from what I've seen of it, the competition had better watch out, too.

Who in their right mind would buy a BBC B+ at £469 – then spend £250 on a monitor and £150 on a disc drive, when a machine with far better facilities and over 10 times the memory can be brought for £100 less! And as for the Sinclair Quality Lapse, well . . .



Getting to grips with the graphics modes

Part Four of DAVE RUSSELL's series on the Atari graphics modes

SO far in this series we've looked at the text modes obtained using Graphics 0, 1 and 2. This month we'll start in on the actual graphics modes, or map modes as they're sometimes called.

Modes 3, 5 and 7 can be conveniently taken together because they are all four-colour modes. They differ in the size of the smallest block, or pixel, that can be placed on the screen. That is, they offer different levels of resolution and therefore make different demands on memory space.

The pixels in a Mode 3 screen are the same size as those of Mode 0. If you type:

GRAPHICS 3

most of your screen will go black and you'll be left with the familiar blue text window at the bottom.

You now have 20 rows of 40 columns on which you can produce your display. Before we start trying to put anything up there, let's get the colour registers sorted out in our minds.

I often think that the designers of the Atari and the writers of the Basic were kept apart in order to produce as many different numbering systems as possible.

I'm not always this cynical – it comes over me when I try to remember all the different schemes for selecting a colour.

Registers 0, 1 and 2 hold the information for the foreground colours and register 4 holds the information for the background colour.

Register 4 defaults to black while registers 0, 1 and 2 default to orange, light green and dark blue respectively.

We can use the SETCOLOR command to alter these colours. With a clear Graphics 3 screen, try typing:

SETCOLOR 4,13,0

The black background should have been replaced by darkish green because you have changed register 4 to colour 13 with luminance of 0.

When we want to put something on the screen, the COLOR command selects which of the registers to take the colour information from.

It's here where the numbering starts getting tied in knots, because COLOR 0 selects the background colour information in register 4.

COLOR 1 selects register 0, COLOR 2 selects register 1 and COLOR 3 selects register 2. As you can see, for these three registers the COLOR number is one more than the register it selects.

It's not difficult, but it could have been simpler.

At this point I suggest you press Reset to get back to default conditions and then type GRAPHICS 3 (or GR.3 to save a bit of typing). Now let's put something up there.

Try typing:

COLOR 1: PLOT 15,15

This should yield an orange square fairly close to the text window.

The COLOR 1 selects the colour in register 0 (orange) and the PLOT 15,15 fills the pixel at screen position 15,15 with a block of this colour.

Press Reset again and try entering Program I. When you Run it, two things should be demonstated. Firstly, the size of the orange block gets smaller as we move from Mode 3 through Mode 5 to Mode 7. In other words, Mode 7 has higher resolution than Modes 3 or 5.

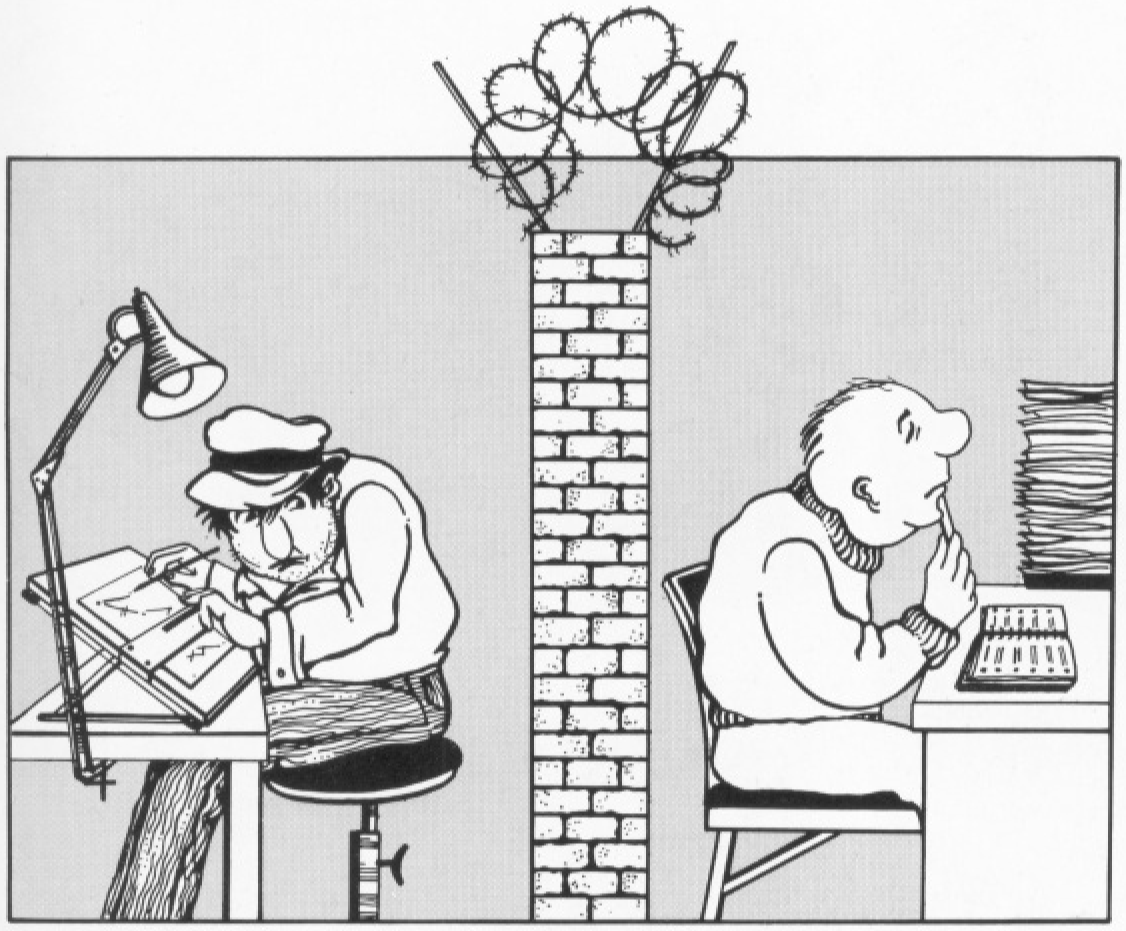
Secondly, the orange square 'moves' up towards the top left-hand corner as the mode changes. This illustrates that screen position 0,0 is actually at the top corner.

```
10 FOR A=3 TO 7 STEP 2
20 GRAPHICS A
30 COLOR 1:PLOT 15,15
40 FOR DELAY=1 TO 300:NEXT DELAY
50 NEXT A
```

Program I

For many people it seems more natural to think of 0,0 as being the bottom left-hand corner. You'll have to remember this or your displays will have a nasty habit of appearing upside-down.

If you've run Program I you'll be



'I often think the designers of the Atari and the writers of the Basic were kept apart in order to produce as many different numbering systems as possible . . .'

left in Mode 7 so press Reset and go back to GRAPHICS 3.

If you enter:

COLOR 1: PLOT 15,15

again you'll once more have the orange square.

Inspirational, isn't it?

In addition to the PLOT command, the other main command for producing displays in these modes is DRAWTO. As you might imagine, this command causes a line to be DRAWn from the last PLOTted position TO the specified position.

If you enter DRAWTO 15,4 a vertical orange line should appear. Try DRAWTO 20,4 to produce a horizontal line.

If you now enter DRAWTO 30,15 you'll see how sloping lines are produced – not very well in a low resolution mode like this, but it's often adequate.

Program II shows how you can produce simple displays using this method. It's all in the same colour, but if you add a line:

25 COLOR 2

you can change the colour used by lines 30 and 40.

```
10 GRAPHICS 3
20 COLOR 1:PLOT 19,5:DRAWTO 19,14
30 PLOT 17,5:DRAWTO 17,11:DRAWTO 14,14
40 PLOT 21,5:DRAWTO 21,11:DRAWTO 24,14
```

Program II

You can also change the mode number in line 10 to either 5 or 7 and see the effect of increasing resolution.

Because Mode 3 has the same pixel size as Mode 0, the bottom right-hand corner of a Mode 3 screen is position 39,19. If you try to PLOT 39,20 you won't see anything happen because the text window is effectively covering row 20.

However, if you try to PLOT 40,19 you'll get an ERROR 141 telling you that the cursor is out of range.

In Mode 5 the bottom right-hand corner is position 79,39 and in Mode 7 it is 159,79. But while Mode 3 needs only 434 bytes of memory, Mode 5 needs 1174 bytes and Mode 7 needs 4190 bytes. As you can see, the extra resolution costs memory.

You don't need to specify a particular screen point in the PLOT command. You can provide PLOT with an expression to evaluate, the result of which will give the position to be plotted.

Program III gives a brief example of this. We can't say where each point will be plotted until the random number generator has been used twice.

You can also print to the screen using the PRINT #6 format that we used in Modes 1 and 2. The only difference is that you can't print an actual character like a letter or number.

```
10 GRAPHICS 3
20 COLOR 1
30 FOR A=1 TO 20
40 PLOT INT(RND(0)*39+1),INT(RND(0)*19
+1)
50 NEXT A
```

Program III

Being a map mode, pixels are either lit (in a colour) or unlit (in the background colour).

To demonstrate the effect, press Reset and enter GR.3 to get a clear Mode 3 screen. Now type:

POSITION 15,15:PRINT#6;"1"

and you should see the orange square that we started from. Try substituting a 2 or a 3 for the 1 and see the effect.

There are times when it's easier to use the PRINT #6 than PLOT and DRAWTO. Program IV produces a chunky Mode 3 display using a combination of the two methods.

It's not brilliant programming but it might give you some ideas while you're hanging around waiting for the postman to deliver the next issue of *Atari User*.

```
10 DIM A$(3),B$(5),C$(2)
20 A$="111":B$="22222":C$="33"
30 GRAPHICS 3
40 COLOR 3:PLOT 10,17
50 DRAWTO 10,1:DRAWTO 19,1:DRAWTO 19,3
60 FOR A=4 TO 6
70 POSITION 18,A:PRINT #6;A$
80 NEXT A
90 FOR A=7 TO 10
100 POSITION 17,A:PRINT #6;B$
110 NEXT A
120 FOR A=11 TO 14
130 POSITION 17,A:PRINT #6;C$;" ";C$
140 NEXT A
150 POSITION 14,7:PRINT #6;A$
160 POSITION 22,7:PRINT #6;A$
170 POSITION 16,15:PRINT #6;A$;" ";A$
```

Program IV

How much it costs to use MicroLink

Initial registration fee: £5.

Standing charge: £3 per calendar month or part.

Connect charge: 3.5p per minute or part – cheap rate; 10.5p per minute or part – standard rate.
Applicable for duration of connection to the Service. Minimum charge: 1 minute.
Cheap rate is from 7pm to 8am, Monday to Friday, all day Saturday and Sunday and public holidays; Standard rate is from 8am to 7pm, Monday to Friday, excluding public holidays.

Filing charge: 20p per unit of 2,048 characters per month.
Applicable for storage of information, such a telex, short codes and mail files. The number of units used is an average calculated by reference to a daily sample.

Information Databases: Various charges.
Any charges that may be applicable are shown to you before you obtain access to the database.

MicroLink PSS service: 2p per minute or part (300 baud); 2.5p per minute or part (1200/75 baud).
Only applies to users outside the 01- London call area.

Telex registration: £10.

Outgoing telex: 5.5p per 100 characters (UK); 11p per 100 (Europe); 16.5p per 100 (N. America); £1.15 per 400 (Rest of world); £2.75 per 400 (Ships at sea).
Deferred messages sent on the night service are subject to a 10 per cent discount.

Incoming telex: 50p for each correctly addressed telex delivered to your mailbox. Obtaining a mailbox reference from the sender incurs a further charge of 50p.
It is not possible to deliver a telex without a mailbox reference. If a telex is received without a mailbox reference the sender will be advised of non-delivery and asked to provide a mailbox address.
Each user validated for telex and using the facility will incur a charge of 6 storage units a month. Further storage charges could be incurred depending on the amount of telex storage and the use made of short code and message file facilities.

Telemessages: £1.25 for up to 350 words.

Radiopaging: No charge.
If you have a BT Radiopager you can be paged automatically whenever a message is waiting in your mailbox.

International Mail: For the first 2,048 characters – 20p to Germany and Denmark; 30p to USA, Australia, Canada, Singapore, Hong Kong and Israel. For additional 1,024 characters – 10p; 15p.
These charges relate to the transmission of information by the Dialcom service to other Dialcom services outside the UK and the Isle of Man. Multiple copies to addresses on the same system host incur only one transmission charge.

Billing and Payment: All charges quoted are exclusive of VAT. Currently all bills are rendered monthly.

Software over the telephone

MicroLink is setting up a central store of software programs which you'll be able to download directly into your micro. The range will include games, utilities, educational and business programs, and will cover all the most popular makes of micros.

Talk to the world – by satellite

MicroLink is part of the international Dialcom network. In the USA, Australia and a growing number of other countries there are many thousands of users with electronic mailboxes just like yours. You can contact them just as easily as you do users in Britain – the only difference is that the messages from your keyboard go speeding around the world via satellite.

What you need to access MicroLink

You must have three things in order to use MicroLink: a computer (it can be any make of micro, hand-held device or even an electronic typewriter provided it has communications facilities), a modem (it can be a simple Prestel type using 1200/75 baud, or a more sophisticated one operating at 300/300 or 1200/1200 baud), and appropriate communications software.

MicroLink
in association with
TELECOM GOLD *
Application Form

I/We hereby apply to join MicroLink

(✓) □ I enclose my cheque for £5 payable to Database Publications as registration fee to MicroLink.

(✓) □ I also wish to use Telex. I authorise you to charge an additional £10 to my initial bill for validation.

□ I confirm that I am over 18 years of age.

Signature ______________________

Date ______________________

I intend to use the following computer ______________________

FOR OFFICE USE ONLY:
Mailbox assigned ______________________
Start date ______________________
Password ______________________

SEND TO:
MicroLink
Database Publications
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY.

Name
Position
Company
Address

Postcode　　　　Daytime telephone

Commencement of Service
Please indicate month of commencement
Allow 10 days for validation of mailbox　　　□□ 19 □□

Payment
Whilst Database Publications Ltd is the supplier of all the services to you, the commission and billing thereof will be handled by Telecom Gold as agents for Database Publications Ltd.
Date of first payment to be on 15th of month following commencement.
Please complete billing authorisation form A, B or C below:

A. Direct Debiting Mandate *(Enter full postal address of Bank Branch)*
To

I/We authorise you until further notice in writing to charge to my/our account with you on or immediately after 15th day of each month unspecified amounts which may be debited thereto at the instance of British Telecommunications plc – TELECOM GOLD by Direct Debit. Bills are issued 10 days before debit is processed.

Name of Account to be debited

Account Number

B. Please debit my/our Access/Visa account number

I/We authorise you until further notice in writing to charge to my/our account with you on or immediately after 15th day of each month unspecified amounts which may be debited thereto at the instance of British Telecommunications plc – TELECOM GOLD. Bills are issued 10 days before charge is applied to your account.

C. Please invoice the company/authority.

(✓) □ If you select this option, which is ONLY AVAILABLE to government establishments and public limited companies, you will be sent an authorisation form for completion which will require an official order number to accept unspecified amounts.

* *Telecom Gold is a trademark of British Telecommunications plc.*

A money-saving special offer from ATARI USER



Price includes VAT, post & packing

THIS is your passport into the exciting world of telecomputing!

IT'S by far the fastest growing field in micro-computing. All over the world micros are talking to each other over the telephone line. As well as to the ever-increasing number of public and private databases, bulletin boards – and even giant mainframe computers.

We want all *Atari User* readers to share in the new technology that makes all this possible. So we're offering a unique starter pack at an unbeatable price. It gives you everything you need to get in touch with the big wide world outside:

- **Modem**
- **Software**
- **Serial Interface**

The modem is the amazing Miracle Technology WS2000. One of the most powerful on the market, it provides all the facilities you require. Yet it's simplicity itself to use. Just plug it into a standard British Telecom jack and you're away!

The package also features the superb Datari serial interface, which links the modem directly to the Atari's peripheral port without the need for the 850 serial interface.

The best hardware deserves the best software to drive it, and with the specially written Multi-Viewterm program the package is complete. It supports all the standard baud rates – 1200/75, 75/1200, 1200/1200 and 300/300 full duplex.

Your *Atari User* package will allow you to talk directly to other computers, to send your own telex messages, to go tele-shopping – even to download free software programs directly into your Atari.

You will be able to join Micronet/Prestel, which will immediately open up to you a vast menu of 750,000 pages of information – instant world news, sports, holidays, hotels, train and airline timetables, all regularly updated.

And you can become one of a growing number of enthusiasts who are joining MicroLink, the giant database set up in conjunction with Telecom Gold, which is described more fully in this issue.

But first, send for the Miracle package – and enter the fascinating, limitless world of communications!

Use the order form on Page 61

Don't you think you need a little protection?

ANDRE WILLEY makes you an offer you can't refuse . . .



ONE of the main problems with Basic, apart from its poor speed in comparison with machine code, is that once you've finished your masterpiece anyone can LIST it to the screen or printer and copy your ideas.

We've had a number of letters asking if there are any ways to prevent someone from pressing Break or System-Reset and LISTing the program, and luckily there are quite a few things you can do.

Let's take the points in order:

Break is perhaps the easiest of all to protect from. It involves just two POKE instructions:

POKE 16,64
POKE 53774,64

To switch the Break key back on again, type:

POKE 16,192
POKE 53774,192

Unfortunately the GRAPHICS command will return Break to its normal use, so you must re-POKE the values after each GRAPHICS statement in your program. A simple GOSUB to a subroutine is probably best.

System-Reset is far harder to protect because it was designed as an all purpose "get-out" key in case your program goes wrong. As such it should function correctly regardless of whatever you have managed to type in.

Luckily there is *one* way to "capture" the Reset key. One of the functions of Reset is to check that DOS or any cassette loaded program is still working correctly.

When a boot cassette or DOS disc loads in it will set three locations in memory to tell Reset what to do to re-initialise the program just loaded.

Location 9 will contain either a 1, for a disc program, or a 2 for cassette. If it contains 0 then no program was booted.

The other two locations are used to tell the computer the address in memory of a small machine code routine to handle the job of checking the main booted program.

These locations are different for cassette and disc, but we will use the cassette ones, 2 and 3, as they are simpler.

So in order to trap System-Reset we must first POKE location 9 with 2 – for cassette boot – and locations 2 and 3 with the address of a machine code routine?

What? You mean that some of you aren't machine-code programmers? Okay, let's cheat.

Basic itself is really just one massive machine code program. Normally you never have to think of it as such, because it is designed in such a way that you never really notice how it works.

If we could find a suitable section of Basic to "borrow", we wouldn't have to write any machine code ourselves.

The obvious routine to use would be RUN, so that the program would simply re-start if you pushed Reset.

But that may not be what you wanted. You may want your program

```
10 REM PROGRAM LISTING 1
20 REM PROTECTS PROGRAM FROM BREAK AND
       SYSTEM-RESET
30 REM VERSION FOR ATARI XL/XE
       COMPUTERS
40 REM FOR ATARI 400/800 COMPUTERS,
       CHANGE LINE 120 TO POKE 2,64
50 REM
100 REM PROTECT FROM SYSTEM RESET
110 TRAP 500
120 POKE 2,52
130 POKE 3,185
140 POKE 9,2
200 POKE 622,255:REM DOES FINE-SCROLL
       ON XL/XE MACHINES
210 GRAPHICS 0:GOSUB 1000:REM "BREAK"
       KEY...
250 PRINT "    PROGRAM RUNNING NORMALL
```

Program I

to go off and do something else rather than start from scratch.

Fine, let's use the GOTO statement then, but how to tell the computer where to go to? Better still, let's use the TRAP command.

If we can convince Basic that an error has occurred after pushing Reset it will jump to a TRAPped line, which may, for instance, disable the Break key again. For instance, if you have typed:

TRAP 500

the program would continue at line 500 after pressing System-Reset. So where inside Basic is the TRAP handling routine?

Atari has so far released three revisions of its Basic, called, with great inspiration A, B and C.

Version A was shipped in cartridge form with all UK Atari 400 and 800 machines. There were a few very minor problems with it so the new 600XL and 800XL machines had Revision B Basic built in.

Unfortunately one or two new bugs crept in to this one also, so Revision C was born. Available on cartridge for £9.95, this Basic is also built into the currect XE range of computers.

The TRAP routine on Rev. A was located at 47424 ($B940), and on Revs. B and C at 47412 ($B934). Thus you must POKE locations 2 and 3 with the correct values.

For Basic Rev. A – cartridge:

POKE 2,64
POKE 3,185

For Basic Revs. B and C – XL/XE range;

POKE 2,52
POKE 3,185

Don't, incidentally, forget to POKE 9,2 as well.

Program I shows Break and Reset protection in use.

This method will disable DOS after Reset is pushed. If you are a disc user and you wish to re-enable DOS, type POKE 9,1 and push System-Reset. The system should then be returned to normal.

There are some rather nice little things you can do to stop your program being LISTed if it has been loaded but not RUN.

The first is to scramble any variable names so that garbage is printed out instead. Program II will do this for you.

It should be typed in on a spare program line, say 32000, run with a GOTO statement, and then deleted. Don't forget to save an original version because even *you* won't be able to read or alter your program once it's been scrambled.

Without going into too much technical detail, for which see "The Atari Basic Source Book", or "Mapping the Atari", both from Compute! Books, it works by putting a Return character instead of each variable name in the listing of the program, thus making it a little tricky to read.

Program III is even more dramatic. This one won't allow *any* commands to be typed in after the routine has been run, hence the SAVE command *must* be in the running portion of the program or you've lost it forever.

This also means that you can't LOAD, or CLOAD, then RUN the program. You must RUN C: or RUN D:Filename. Ext.

Again, I won't go into technical details, but this version will make Basic fail to recognise any lines, either program or command, that you subsequently type in. It effectively forgets where to store them.

Drastic, but quite effective.

One last tip to play about with. Try this:

POKE 202,1

Put it as the first line of the program, and check that it is correct by LISTing it. Try listing it again after you've RUN the program.

You'd better save the program before running that last one. Have fun.

WHICH version of Basic have you got? If you have Basic Rev. A, typing PRINT PEEK(47424) will give a result of 169.

If you have Basic Rev. B or C, typing PRINT PEEK(47424) will give a result of 133, but PRINT PEEK(47412) will print 169.

Any other results from these PEEKs and the Reset protection routine will almost certainly not work.

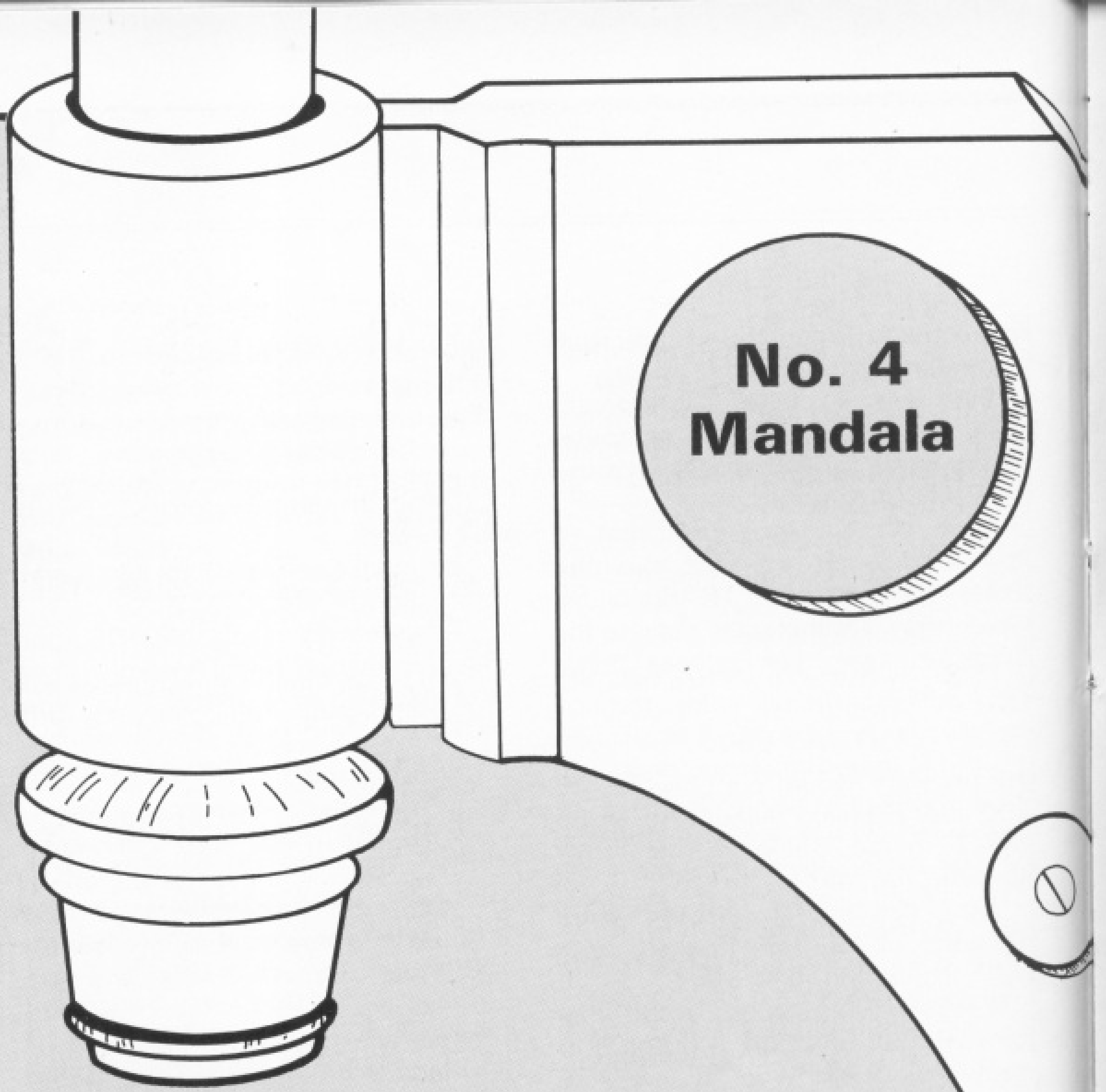
```
Y..."
260 PRINT :PRINT "  TRY PRESSING BREA
K OR RESET ";
270 FOR I=1 TO 500:NEXT I
280 ? :? :? :? :GOTO 250
500 REM COMES HERE IF RESET PRESSED
510 GOSUB 1000:REM "BREAK" KEY...
520 TRAP 500:REM RE-SET TRAP LINE
530 POSITION 9,10:PRINT "RESET HAS BEE
N PRESSED"
540 FOR I=1 TO 1000:NEXT I
550 GOTO 200
1000 REM PROTECT FROM "BREAK" KEY
1010 POKE 16,64:POKE 53774,64
1020 POKE 752,1:REM JUST TO TURN OFF
     CURSOR
1030 RETURN
```

```
10 REM PROGRAM LISTING 2
20 REM SCRAMBLE VARIABLE NAMES
30 REM DON'T FORGET TO SAVE YOUR
     PROGRAM FIRST
40 REM TYPE IN YOUR PROGRAM THEN TYPE
     THE LINES BELOW
50 REM TO RUN, TYPE:  GOTO 32000
60 REM
32000 FOR VAR=PEEK(130)+PEEK(131)*256
TO PEEK(132)+PEEK(133)*256
32010 POKE VAR,155:REM OR ANY OTHER
     ASCII CHARACTER YOU WANT...
32020 NEXT VAR
```

Program II

```
10 REM PROGRAM LISTING 3
20 REM LOCK-UP COMMAND ENTRY MODE
30 REM DON'T FORGET TO SAVE YOUR
     PROGRAM FIRST!
40 REM TYPE IN YOUR PROGRAM THEN TYPE
     LINE 32030
50 REM CHANGE SAVE "D:Filename.Ext"
     TO SAVE "C:" FOR CASSETTE.
60 REM DO NOT USE CSAVE !
70 REM TO RUN, TYPE:  GOTO 32030
80 REM CAN BE USED WITH, OR AFTER,
     PROGRAM 2 EXECUTION
90 REM
32030 POKE PEEK(138)+PEEK(139)*256+2,0
:SAVE "D:Filename.Ext":NEW
```

Micro Scope



No. 4 Mandala

MANDALA is an elementary but very effective program that draws a pattern on the screen.

The program itself is very simple, with only 10 active lines. But the logic behind it isn't trivial.

Try working it out with pencil and paper and you'll soon see the pattern emerging.

```
10 REM MANDALA
20 GRAPHICS 8+16
30 COLOR 1:PLOT 150,90
40 FOR X=0 TO 100 STEP INT(RND(0)*10+1
)
50 DRAWTO 150,190-X
60 DRAWTO 150-X,100
70 DRAWTO 150,X
80 DRAWTO 150+X,100
90 NEXT X
100 FOR DELAY=1 TO 750:NEXT DELAY
110 RUN
```

10 A REM containing the program name.

20 Selects full-screen Graphics 8 mode.

30 Selects colour and plots the starting point.

40-90 These lines define a FOR ... NEXT loop which draws the pattern. Each time round the loop four lines are drawn. The changing value of X changes the positions of the lines. The step size is chosen randomly within the range 1-10 so that variations in the pattern density are produced.

100 Delay to keep the display on the screen long enough to be seen.

110 Start again.

68000 POWER TO THE PROGRAMMER

THE 8 bit microprocessors have been around for over 10 years now. While there have been many improvements in hardware in that time the philosophy of processor design has served quite well.

Now that it is possible to get even more circuitry on to a chip, a new breed of 16 and 32 bit microprocessors are emerging which have power that is not so "micro".

Most manufacturers of the new generation of processors were involved with the earlier 8 bit ones, and it is interesting to see how they viewed their development.

One approach is to keep things as similar as possible.

This has the advantage of not requiring vastly new skills, but tends to "freeze in" all the design errors and compromises that were made in the past.

Another approach is to bolt on increasingly more powerful commands giving large raw processing power.

This approach produces very powerful processors that do well in bench mark tests but are rather difficult to bend to your particular application.

In other words, a racing car rather like a dragster, unbeatable in acceleration but a swine on the corners!

The final approach is to look at the code that was written on the 8 bit processors and analyse it for sequences.

Which means, find out what the programmer wants to do and then design a processor that will make it easier for him to do it.

This produces a "Programmer's Processor" and is the approach taken by Motorola in designing the 68000 microprocessor.

While the other approaches have been tried with some success measured in sales, it is rumoured that most programmers working on these other processors have a 68000 as a pinup fantasy on their office walls.

The secret of this lies in the instruction set, the basic commands that all other commands must be made from.

In a word, it is very "orthogonal", which means that you don't have to worry about what commands you can perform with what registers on what memory locations.

If you want to do an operation the odds are that there is an instruction/addressing mode combination to do it.

This will be worked out for you by a good assembler – all you have to do is specify the source and destination of the operation.

While it is possible, and in most cases desirable, to program 8 bit processors in hex, looking up the code for each instruction, this would soon drive you up the wall with the 68000.

There are so many different combinations of addressing mode and instructions that you have to "construct" a machine code instruction from the bit patterns which specify the source and destination locations.

So, in practice it would take you at least 30 seconds to work out each instruction.

Obviously this soon mounts up and becomes totally unacceptable. But with an assembler, however, programming is as sweet as a dream.

MIKE COOK looks at the new breed of microprocessors whose power, he reveals, is anything but "micro'

Let's see exactly what the instructions are which make it such a joy to use.

Of course I cannot hope to do justice to this in a short article, but I hope to be able to give you the flavour of what's available.

Next month we will look at the addressing modes and finally at the hardware structure.

Firstly, the way the memory is organised is in bytes.

Each byte has an address, but, as the data bus is 16 bits wide, the least significant address line is not brought out.

So data is fetched two bytes at a time, known as a word.

All of the internal registers are 32 bits long, which takes four byte addresses or two word locations to store them.

Consequently a 32 bit quantity is referred to as a "long word".

So most instructions can be performed on a byte, a word or a long word.

To simplify matters, all word operations must be performed on even byte addresses.

So, for example, if you want to store a word at address location 4, the most significant byte goes in location 4 and the least significant byte goes in location 5.

This is what I consider to be the right way round as we write the most significant part of a number first. But notice that this is the reverse of the way the 6502 handles numbers.

Now let's look at what registers we

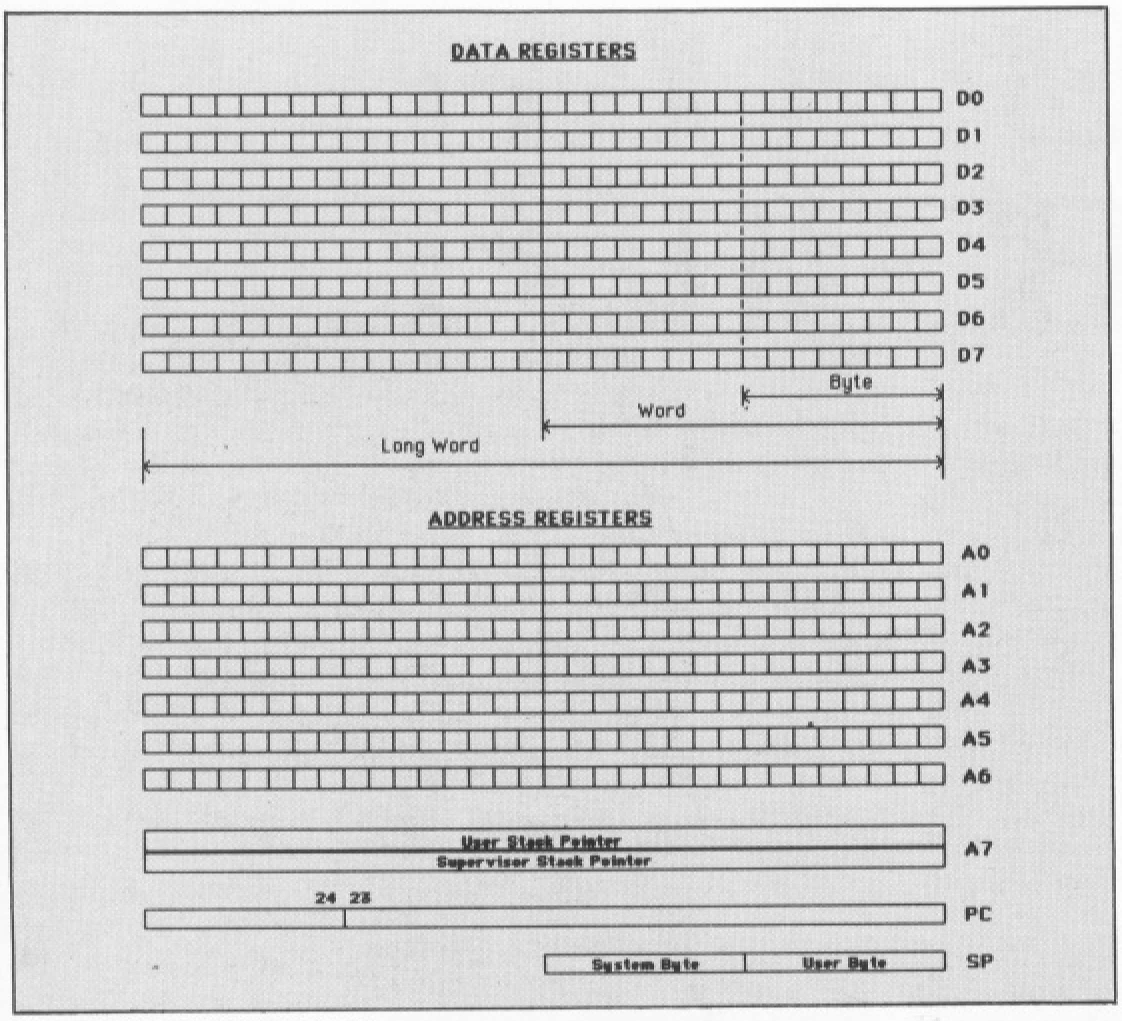


Figure I: The 68000 Registers

have in the 68000. These are shown in Figure I.

As you can see, there are quite a lot of them. The two main types are the Data and Address registers and all of these are 32 bits long.

In general, data registers can be very freely manipulated, and most instructions will operate on them.

Address registers, on the other hand, are mainly used to determine what memory address to use.

Address register 7 (A7) is used as the stack pointer, but any other address register can be so used.

All the subroutine return addresses use the A7 register as their stack pointer, so you can have separate data and return stacks.

This is very useful when implementing high-level languages such as Pascal and Forth.

You may have noticed that register A7 appears to be two registers and so it is.

The 68000 can run in two modes – supervisor and user modes. This means that your operating system can run in the supervisor mode and your application in the user mode.

This makes trace operations easier as well as error handling.

Each mode also has a separate status register.

There is also a program counter. In most microprocessors this is normally the largest register, but paradoxically here it is one of the smallest.

Only the lower three bytes are brought out, thus limiting the memory to 24 mbytes.

In future versions of the chip these extra locations might be brought out, but there is more than enough memory space for the time being.

The most common instructions used in any program are loading and storing of registers. In the 68000 these have been simplified to a single MOVE command.

The source and destination can be quite freely specified to give you exactly the effect you want.

You can even move data between memory locations without passing through any of the registers.

There is even a "move multiple registers" instruction which allows any number of registers to be quickly saved or restored from memory.

Regarding program structure, there are plenty of conditional branch instructions.

There is also an instruction which decrements a register and branches if the register has not yet reached zero. I wish I had a pint for every time I have used that combination!

There are the usual collection of logic operations including shifts.

However, a single instruction can specify any number of shifts to left or right.

The big plus of this class of instructions are the multiply and divide instructions.

When using the multiply instruction, only 16 bits of the registers can be used because the result of two 16 bit operations is a 32 bit value.

There are also instructions which allow the operations to be signed or unsigned.

Another class of instructions are the Trap Instructions. These are like a single instruction call-to-subroutine.

When they are used, the program goes to an address stored in a fixed memory location, and these locations are known as the Trap vectors.

They are very handy for communicating with the operating system in a standard way.

If all input and output is done through these traps, then programs written for one hardware configuration of the 68000 can easily be modified to run on another.

This is very much the way the CP/M operating system works.

These instructions also allow the expansion of the instruction set by providing an easy way to call Macro commands – the Apple Macintosh makes extensive use of these.

Perhaps the newest of instructions are the Link and Unlink. These are capable of implementing a frame pointer to allow an area of memory to be dynamically allocated and de-allocated.

You can use them to store local variables in procedures and to return values when the procedures are finished.

This is vital when procedures are being called recursively.

This feature makes the implementation of Pascal especially easy.

With all these instructions at your command, the task of programming is made very much easier than on any 8 bit processor.

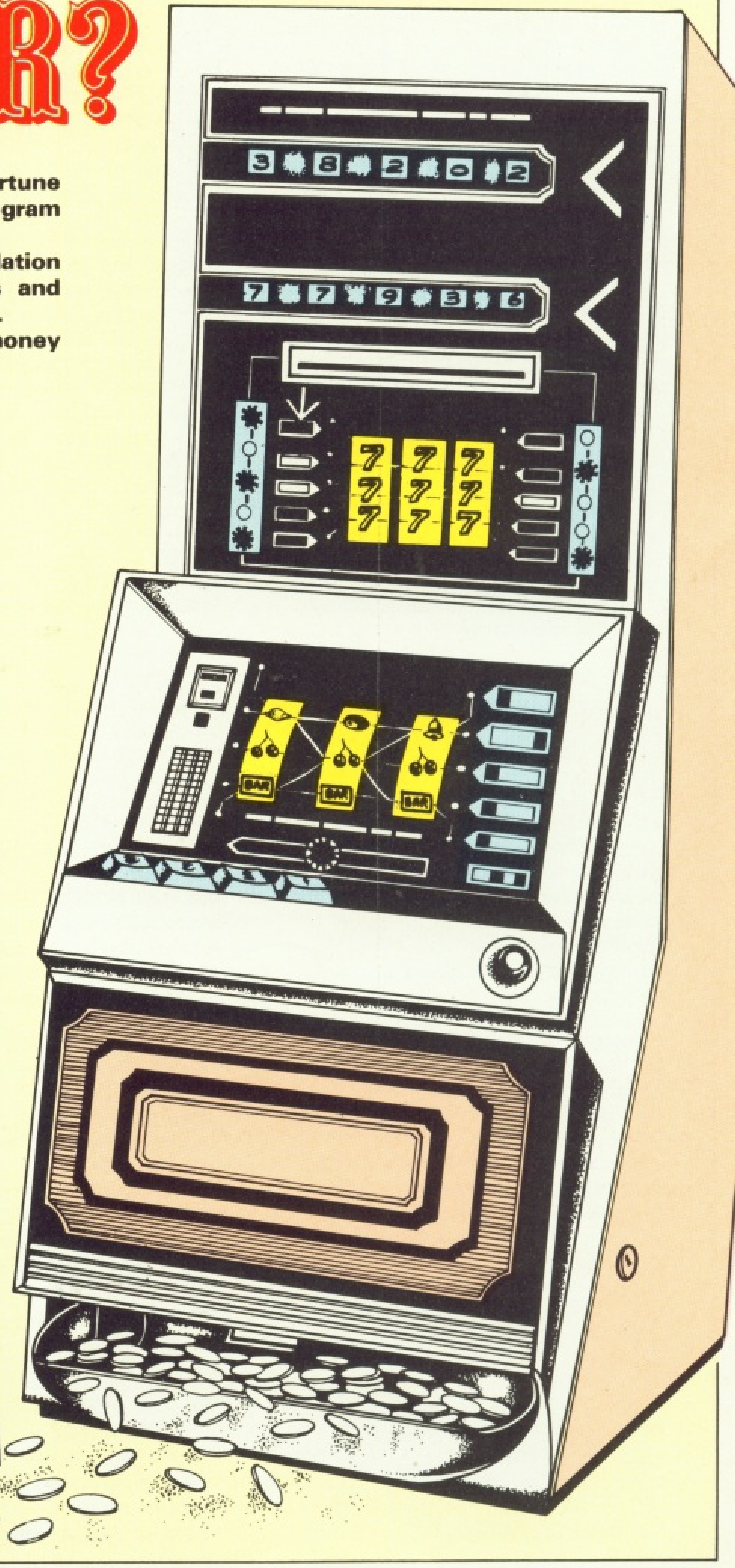
● *Next month we will see how these powerful instructions combine with a multitude of addressing modes to produce a very versatile instruction set.*

ARE YOU A FRUITI GAMBLER?

IF you're one of those people who spend a fortune on the fruit machine in your local, here's a program from CLIVE PALMER to save you money.

Fruiti Gambler is a fruit machine simulation complete with Hold and Nudge features and incorporating a special Gamble feature reel.

While you're typing it in, think of the money you'll save.



PROGRAM STRUCTURE

50-70	GOTO initialise routine.
90-130	Main program loop.
150-260	Print a reel routine.
280-390	Spin reels 1-3 T1 times.
410-660	Nudge routine.
680-900	Test for a win: No Win - Return, :Win - Gamble?.
910-920	Clear message screen.
930-960	How much is left in bank?
970	Decrement position in reels by 1.
980	Increment position in reels by 1. (used by WIN routine.)
990	Turn all sound channels off.
1010-1070	Start.
1090-1220	Hold reel routine.
1240-1600	Set up new character set.
1610-1710	Define/initialise main variables.
1730-1860	Draw screen display.
1880-2120	Initialise display list interrupt.
2140-2190	Re-start/finish routine.
2200-2310	Gamble routine.

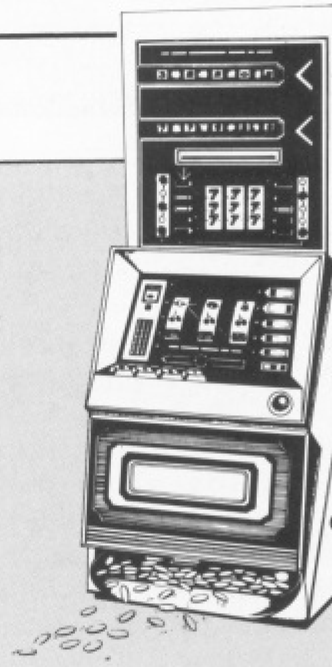
MAJOR VARIABLES

FRUIT$(64)	Contains all fruits for reels.
X(4) & Y(4)	Position of reels.
POS(4)	Pointer showing where we are in reel.
H$(3)	Used to determine if a reel is HELD.
REEL(4,32)	Used to hold reel/fruit data.
WIN(11)	Winning amounts for a winning line.
CASH$(12)	Used to change cash into inverse before displaying on the screen.
CASH	How much you have.
PAY	How much you won.

Game

```
10 REM ********************************
20 REM *      THE FRUITI GAMBLER       *
30 REM *   WRITTEN BY CLIVE PALMER.    *
40 REM ********************************
50 W=0:O=1:P=O+O:Q=P+O:U=Q+O
60 POKE 82,W:GOSUB 1240:POKE 756,GRTOP
:POKE 752,0:CASH=50
70 POSITION W,W:? :? "    FRUITI-GAMBL
ER (C) FOR ATARI 32K."
80 REM *    MAIN LOOP     MAIN LOOP    *
90 H$="...":GOSUB 1010:N=W
100 GOSUB 280
110 GOSUB 680
120 IF N=O THEN GOSUB 410
130 GOTO 90
140 REM *          PRINT REELS          *
150 POS(R)=POS(R)+O
160 IF POS(R)=31 THEN POS(R)=O
170 GOTO 200
180 POS(R)=POS(R)-O
190 IF POS(R)=W THEN POS(R)=30
200 TEMP=REEL(R,POS(R))*U
210 POSITION Y(R),X(R):? FRUIT$(TEMP-Q
,TEMP-P);"↓←←";FRUIT$(TEMP-O,TEMP);
220 TEMP=REEL(R,(POS(R)+O))*U
230 ? "↓←←";FRUIT$(TEMP-Q,TEMP-P);"↓←←
";FRUIT$(TEMP-O,TEMP);
240 TEMP=REEL(R,(POS(R)+P))*U
250 ? "↓←←";FRUIT$(TEMP-Q,TEMP-P);"↓←←
";FRUIT$(TEMP-O,TEMP)
260 RETURN
270 REM *     SPIN REELS T1 TIMES       *
280 T1=INT(RND(W)*50)+O:IF T1<25 THEN
280
290 T2=INT(RND(W)*15)+O
300 T3=INT(RND(W)*10)+O:IF T3>T2 THEN
290
310 R1=T1-T2:R2=T1-T3
320 FOR SPIN=O TO T1
330 FOR R=O TO Q
340 IF H$(R,R)="H" THEN 380
350 IF R=O AND SPIN>R1 THEN 380
360 IF R=P AND SPIN>R2 THEN 380
370 GOSUB 180
380 NEXT R:NEXT SPIN
390 RETURN
400 REM *NUDGE! NUDGE! NUDGE! NUDGE!*
410 POKE 764,255
420 POSITION U+Q,20:? " PRESS START FO
R NUDGES "
430 NUM=INT(RND(W)*12)+O:IF PEEK(53279
)<>6 THEN 430
440 C=W:POSITION U+Q,18:? "***NUDGE DO
WN***"
450 POSITION W,19:? "  USE  OPTION = 1
,SELECT = 2,START = 3       PRESS "U" TO
 NUDGE UP ONLY ........     "
460 IF NUM>10 THEN NUM=30000
470 FOR NUDGE=W TO NUM-O:POSITION 19,Q
+P:? "   ";"←←←";
480 IF NUM>10 THEN ? "UL!":GOTO 500
490 ? NUM-NUDGE
500 POSITION U+U,Q+P:? "NUDGENUDGE=":S
OUND W,250,10,10
510 FOR T3=O TO 20:NEXT T3:POSITION 8,
P+Q:? "NUDGENUDGE=":SOUND W,220,10,10:
FOR T3=O TO 20:NEXT T3
520 T1=PEEK(764):POKE 764,255
530 T2=PEEK(53279)
540 IF T1=11 THEN POSITION 16,18:? " U
P *"
550 IF T1<>11 THEN POSITION 16,18:? "D
OWN*"
560 IF T2=5 THEN R=P:GOTO 610
570 IF T2=6 THEN R=Q:GOTO 610
580 IF T2=Q THEN R=O:GOTO 610
590 C=C+O:IF C=50 THEN C=W:GOTO 650
600 GOTO 500
610 C=W:POKE 53279,8:GOSUB 990
620 IF T1=11 THEN GOSUB 150:GOTO 640
630 GOSUB 180
640 GOSUB 680
650 NEXT NUDGE
660 GOSUB 910:GOTO 780
670 REM *  HAVE YOU WON ANYTHING ?   *
680 GOSUB 980
690 IF REEL(O,POS(O)+O)=O AND REEL(P,P
OS(P))=O AND REEL(Q,POS(Q)-O)=O THEN P
AY=WIN(REEL(P,POS(P))):GOTO 810
700 IF REEL(O,POS(O)-O)=O AND REEL(P,P
OS(P))=O AND REEL(Q,POS(Q)+O)=O THEN P
AY=WIN(REEL(P,POS(P))):GOTO 810
710 IF REEL(O,POS(O))=REEL(P,POS(P)) A
ND REEL(Q,POS(Q))=REEL(P,POS(P)) THEN
PAY=WIN(REEL(O,POS(O))):GOTO 810
720 IF REEL(O,POS(O))=REEL(P,POS(P)) A
ND REEL(Q,POS(Q))<>REEL(P,POS(P)) THEN
 PAY=20:GOTO 810
730 IF REEL(O,POS(O))=8 AND REEL(P,POS
(P))<>8 AND REEL(Q,POS(Q))<>8 THEN PAY
=10:GOTO 810
740 IF REEL(O,POS(O))=8 AND REEL(P,POS
(P))=8 AND REEL(Q,POS(Q))<>8 THEN PAY=
15:GOTO 810
750 IF REEL(P,POS(P))=11 THEN PAY=WIN(
REEL(P,POS(P))):GOTO 810
760 GOSUB 970:IF N=O THEN RETURN
770 IF N=W AND (INT(RND(W)*10)+O)<Q TH
EN N=1:RETURN
780 FOR T1=127 TO 200:SOUND W,T1,10,10
:SOUND O,T1-10,10,10:SOUND P,T1-20,10,
10:SOUND Q,T1-30,10,10:NEXT T1
790 GOSUB 990
800 RETURN
810 GOSUB 970:GOSUB 910:IF PAY<100 THE
N GOSUB 2200
820 IF PAY=W AND N=O THEN GOSUB 910:PO
SITION U+U+P,20:? "HARD LUCK!":POP :GO
TO 780
830 IF PAY=W THEN GOSUB 910:POSITION U
+U+P,20:? "HARD LUCK!":GOTO 780
840 FOR T1=127 TO 0 STEP -O:SOUND W,T1
,10,10:SOUND O,T1+10,10,10:SOUND P,T1+
20,10,10:SOUND Q,T1+30,10,10:NEXT T1
850 GOSUB 990
860 GOSUB 910:POSITION 7,20:? "COLLECT
 ";PAY;"P";"  PRESS  START "
870 IF PEEK(53279)<>6 THEN 870
880 CASH=CASH+PAY:GOSUB 930:GOSUB 910
890 IF N=O THEN POP
900 RETURN
910 POSITION W,19:? "


                     ";
920 ? "
        ":RETURN
930 POSITION U+Q,18:? "*BANK*";:CASH$=
"£":CASH$(P)=STR$(CASH/100):X=LEN(CASH
$):CASH$(X+1)="0"
940 FOR T1=X+2 TO Q*Q:CASH$(T1)="*":NE
XT T1:X=LEN(CASH$)
950 FOR T1=O TO X:? CHR$((ASC(CASH$(T1
,T1))+128);
960 NEXT T1:RETURN
970 FOR X=O TO Q:POS(X)=POS(X)-O:NEXT
X:RETURN
980 FOR X=O TO Q:POS(X)=POS(X)+O:NEXT
X:RETURN
990 FOR X=W TO Q:SOUND X,W,W,W:NEXT X:
RETURN
1000 REM *   STARTING SCREEN DISPLAY   *
1010 POSITION U+U,16:? "  →→  →→  "
1020 CASH=CASH-10:IF CASH<W THEN 2140
1030 GOSUB 930
1040 IF INT(RND(W)*10)+O>U+U THEN 1090
1050 POSITION W,20:? "      PRESS  STAR
T  TO SPIN REELS       "
1060 IF PEEK(53279)<>6 THEN 1060
1070 GOSUB 910:POKE 77,W:POSITION 8,U+
O:? "FRUITI-GAMBLER":RETURN
1080 REM * HOLD HOLD HOLD HOLD HOLD *
1090 POKE 764,255:H$="..."
1100 POSITION W,19:? "  USE  OPTION =
1,SELECT = 2,START = 3  TO HOLD REELS,
C = CANCEL,S = SPIN REELS."
1110 FOR T3=O TO 50:NEXT T3
1120 POSITION U+U,U+O:? "HOLD HOLD HOL
D":T2=PEEK(53279)
1130 T1=PEEK(764):POKE 764,255
1140 IF T1=18 THEN GOSUB 1220
1150 IF T2=Q THEN H$(O,O)="H":POSITION
 8,16:? "▼"
1160 IF T2=5 THEN H$(P,P)="H":POSITION
 12,16:? "▼"
1170 IF T2=6 THEN H$(Q,Q)="H":POSITION
 16,16:? "▼"
1180 SOUND W,220,10,10
1190 FOR T3=O TO 15:NEXT T3:GOSUB 990
1200 IF T1<>62 THEN POSITION U+U,U+O:?
 "HOLD HOLD HOLD":GOTO 1110
1210 GOSUB 910:GOTO 1070
1220 POSITION 8,16:? "  →→  →→  ":GOTO
 1090
1230 REM *    CHANGE CHARACTER SET     *
1240 RESTORE 1320
1250 MEMTOP=PEEK(106):GRTOP=MEMTOP-U:P
OKE 106,GRTOP-U:GRAPHICS W:CHROM=PEEK(
756)*256:CHRAM=GRTOP*256
1260 SETCOLOR U,8,W:SETCOLOR P,8,W:SET
COLOR O,W,14:POKE 752,1
1270 POSITION U,U:? "  "FRUITI-GAMBLER"
 FOR ATARI 32K "
1280 ? "↓↓↓↓↓↓↓↓SETTING UP ... PLEASE
WAIT"
1290 POKE 756,GRTOP:FOR A=W TO 1023:PO
KE CHRAM+A,PEEK(CHROM+A):NEXT A
1300 FOR K=O TO 64:READ CH
1310 FOR I=W TO 7:READ A:B=I+CH*8+256*
GRTOP:POKE B,A:NEXT I:NEXT K
1320 DATA 97,1,2,1,7,7,15,15,15,98,128
,64,128,224,224,240,240,240,99,31,31,6
3,63,127,127,1,1,100,248,248,252,252
1330 DATA 254,254,128,128
1340 DATA 101,0,255,255,0,113,74,74,11
5,102,0,255,255,0,156,82,82,220,103,74
,74,74,114,0,255,255,0
1350 DATA 104,88,84,82,82,0,255,255,0
1360 DATA 105,0,15,48,71,135,128,129,1
31,106,0,240,12,226,225,225,193,129,10
7,131,131,131,131,64,48,15,0
```



```
1370 DATA 108,1,1,1,1,2,12,240,0
1380 DATA 109,0,0,0,0,0,0,0,0,110,3,7,
15,31,31,63,127,126,111,0,1,3,15,31,12
7,255,252
1390 DATA 112,254,252,252,248,240,224,
128,0
1400 DATA 113,0,0,0,0,1,31,63,127,114,
12,6,118,252,252,254,254,254,115,255,2
55,255,255,255,127,127,31
1410 DATA 116,252,248,248,248,248,240,
224,128
1420 DATA 117,24,28,13,3,61,103,79,207
,118,48,112,224,136,188,254,254,255,11
9,231,255,255,255,255,127,127,31
1430 DATA 120,255,255,255,255,255,254,
254,248
1440 DATA 70,0,3,7,12,24,24,24,63,71,0
,240,240,0,0,0,0,128,72,63,24,24,24,24
,127,127,0,73,128,0,0,0,0,240,240,0
1450 DATA 3,0,12,14,6,1,1,2,4,4,0,24,3
2,88,156,140,192,32,5,8,28,54,111,119,
62,28,0
1460 DATA 6,8,28,62,127,127,62,28,0
1470 DATA 121,0,0,7,30,63,63,127,127,1
22,0,254,159,63,255,255,255,255,11,127
,255,255,255,255,255,127,0
1480 DATA 13,255,254,254,252,248,224,1
28,0
1490 DATA 27,0,255,255,0,33,98,36,36,2
8,0,255,255,0,128,64,44,42,30,36,36,34
,113,0,255,255,0
1500 DATA 32,42,44,72,136,0,255,255,0
1510 DATA 74,0,166,169,233,169,169,166
,0,75,0,70,69,69,69,69,118,0,76,0,0,0,
0,0,0,63,63,77,0,0,128,192,224
1520 DATA 240,248,252,78,63,63,0,0,0,0
,0,0,79,252,248,240,224,192,128,0,0
1530 DATA 80,0,0,1,3,7,15,31,63,83,0,0
,0,0,0,0,252,252,84,63,31,15,7,3,1,0,0
,86,252,252,0,0,0,0,0,0
1540 DATA 85,255,255,255,255,0,0,0,0
1550 DATA 59,6,1,0,15,61,127,118,127,6
0,96,96,128,240,188,254,218,254,61,123
,111,125,54,31,11,7,3
1560 DATA 62,110,250,182,220,248,112,2
24,192
1570 DATA 65,0,255,255,0,0,140,146,146
,66,0,255,255,0,0,103,132,103,87,146,1
46,236,0,0,255,255,0
1580 DATA 88,20,20,231,0,0,255,255,0
1590 DATA 26,0,255,255,0,0,99,54,28,7,
0,255,255,0,0,48,108,12,68,28,54,99,0,
0,255,255,0,15,24,48,124,0,0,255,255,0
1600 DATA 123,24,108,96,120,96,96,252,
0
1610 DIM FRUIT$(64),X(U),Y(U),POS(U),H
$(Q),REEL(U,32),WIN(11),CASH$(U+U+U)
1620 REM *         REEL DATA.          *
1630 DATA 10,7,5,9,3,1,8,4,2,6,10,1,3,
8,9,5,6,7,1,4,8,10,7,9,6,5,8,2,1,4,10,
7
1640 DATA 8,6,4,2,10,1,11,7,3,5,9,10,7
,8,1,3,11,6,9,4,2,8,10,11,6,1,3,7,4,5,
8,6
1650 DATA 2,8,10,6,1,3,7,5,4,9,10,8,1,
7,6,2,5,3,9,8,1,10,2,4,7,5,3,1,6,9,2,8
1660 DATA 12,13,12,14,12,15,12,16,12,1
3,12,14,12,15,12,16,12,13,12,14,12,15,
12,16,12,13,12,14,12,15,12,13
1670 REM *      WIN PRICES DATA.       *
1680 DATA 50,200,150,70,40,50,100,30,4
0,60,10
1690 FOR X=0 TO U:POS(X)=INT(RND(W)*30
)+0:FOR Y=0 TO 32:READ Z:REEL(X,Y)=Z:N
EXT Y:NEXT X
1700 FOR X=0 TO 11:READ Z:WIN(X)=Z:NEX
T X
1710 FRUIT$="abcdefghijklmnopqrstuvwx/
\▄#$%&[\]^yz+-;<>@├ ┤┘*=*1**0:'┤/5**
0":GOSUB 1880
1720 REM *     DRAW FRUIT MACHINE      *
1730 ? "K":GOSUB 1800:POSITION 7,7:? "
┌▀▀─▀▀─▀▀─┐":POSITION 7,14:? "└─
▄▄─▄▄─▄▄─┘"
1740 FOR Z=0 TO 6:POSITION 7,7+Z:? "▌
 █  █  █  ▐":NEXT Z
1750 POSITION 0,10:? "WIN ▄▀":POSITION
 23,10:? "▀▄":POSITION 0,11:? "LINE▀▌"
:POSITION 23,11:? "▐▌"
1760 POSITION 7,15:? "┌▀▀─▀▀─▀▀─▀▀┐
":POSITION 7,16:? "▌ █  █  █  ▐":P
OSITION 7,17:? "└▄▄─▄▄─▄▄─▄▄┘"
1770 POSITION 7,U:? "┌──────────┐"
:POSITION 7,5:? "▌*************▐":POS
ITION 7,6:? "└▄▄▄▄▄▄▄▄▄▄┘"
1780 X(0)=P+Q+Q:Y(0)=X(0):X(P)=X(0):X(
Q)=X(0):Y(P)=P*(Q+Q):Y(Q)=Y(0)*P:Y(P+P
)=Y(0)+Y(P):X(P+P)=X(0)
1790 FOR R=0 TO U:GOSUB 200:NEXT R:GOS
UB 1070:RETURN
1800 POSITION W,0:? "************ ONLY 1
0P ******WINNING LINES*+++++++++++++#$
__ __↓+++++++++%&↓++";
1810 ? "#$ #$ __↓++++++++++%& %&↓+++++";
1820 ? "#$ #$ #$↓++++++++++%& %& %&↓+++++
++++";
1830 ? "ab __ ab↓++++++++++cd    cd↓++++
++++__ ab __↓+++++++++   cd↓+++++";
1840 ? "ab __ ab↓++++++++++cd    cd↓++++
++++__ ;( __↓+++++++++   )e"
1850 POSITION W,18:? "******************
***********OR ANY 2 OR 3*"
1860 RETURN
1870 REM * GRAPHICS 0 DISPLAY LIST. *
1880 RESTORE 2030:FOR N=W TO 99:READ X
:POKE 1664+N,X:NEXT N
1890 COLTAB=1712:LUTAB=COLTAB+24
1900 X=USR(1693)
1910 POKE 512,128
1920 POKE 513,U+P
1930 DSTART=PEEK(560)+256*PEEK(561)
1940 FOR N=DSTART+6 TO DSTART+28
1950 POKE N,130
1960 NEXT N
1970 POKE DSTART+Q,194
1980 POKE 54286,192
1990 PRINT CHR$(125)
2000 POKE 710,PEEK(COLTAB)
2010 POKE 709,PEEK(LUMTAB)
2020 RETURN
2030 DATA 72,138,72,174,156,6,189,176,
6,141
2040 DATA 10,212,141,24,208,189,200,6,
141,23
2050 DATA 208,238,156,6,104,170,104,64
,1,104
2060 DATA 169,7,160,168,162,6,32,92,22
8,96
2070 DATA 169,1,141,156,6,76,98,228,12
8,128
2080 DATA 128,14,206,30,206,252,252,25
2,30,30
2090 DATA 252,252,252,206,30,206,14,12
8,128,128
2100 DATA 128,128,14,14,14,0,0,0,0,0
2110 DATA 0,0,0,0,0,0,0,0,0,0
2120 DATA 0,14,14,14,14,14,0,0,0,0
2130 REM *RESTART  RESTART  RESTART *
2140 POSITION P,20:? " OUT OF LUCK AGA
IN.GO SEE YOUR BANK        MANAGER.I WI
LL BE WAITING !!!!!!"
2150 FOR T1=0 TO 500:NEXT T1:GOSUB 910
2160 POSITION 0,20:? "  SELECT FOR ANO
THER GO,START = QUIT. "
2170 IF PEEK(53279)=6 THEN GRAPHICS W:
POKE 82,P:NEW
2180 IF PEEK(53279)=5 THEN CASH=100:GO
TO 70
2190 GOTO 2170
2200 POKE 764,255:POSITION U+Q,20:? "C
OLLECT OR GAMBLE ";PAY;"P ?"
2210 T2=INT(RND(W)*25)+25
2220 T1=PEEK(764):POKE 764,255:IF T1=1
8 THEN RETURN
2230 IF T1=61 THEN 2250
2240 GOTO 2210
2250 POSITION U+Q,20:? "  GAMBLING ";P
AY;"P          "
2260 FOR SPIN=0 TO T2:R=U:GOSUB 180:NE
XT SPIN
2270 IF REEL(R,POS(R)+0)=12 THEN PAY=P
AY-PAY:RETURN
2280 IF REEL(R,POS(R)+0)=13 THEN PAY=P
AY+100:RETURN
2290 IF REEL(R,POS(R)+0)=14 THEN PAY=P
AY+10:RETURN
2300 IF REEL(R,POS(R)+0)=15 THEN PAY=P
AY*P:RETURN
2310 PAY=PAY+50:RETURN
```

Learn how to talk to your micro

says Brillig

ONE of the first major problems to face the unwary adventurer when he takes his first tentative steps in an alternative world is that although the game itself communicates his surroundings in perfect English, on entering his first command the machine seems incapable of understanding it.

Suppose you find yourself "in a large room, with a door to the west. A book lies open on the table". This is a hypothetical example of how an adventure may begin, and yet depending on the sentence analyser – or parser to the more technical – the responses to instructions given by the player may seem totally incomprehensible.

Let's assume we are dealing with an inquisitive novice adventurer. "Read the book", he types. "I don't understand", says the screen – or something equally helpful such as "Eh...".

Rule one, Mr Novice, is that most adventures will not understand full sentences, but work on the trusty verb/noun principle. Hence READ BOOK will produce the required response, while READ THE BOOK or anything more elaborate will get the computer's equivalent of a puzzled stare.

So lesson one absorbed, Novice tries again. READ BOOK. "You don't have it". This is getting ridiculous thinks Novice, and hurls the cassette case at the cat.

Rule two, as Novice has just discovered, is that usually to do something with an object found in the game it has to be something you possess. Simply being in the same room isn't enough, even if, as in this case, it ought to be.

Remembering all he has learnt so far Novice tries again. GET BOOK. "O.K. You have it". READ BOOK.

At this point Novice will doubtless receive some vital information about his mission, or possibly a crude plug for another game, maybe a bad joke or perhaps a small clue. Anyway, it is at least a glimmer of progress, and the adventure can continue.

Seeing as there is nothing more of interest for our intrepid hero, Novice decides to leave the room. Easier said than done. GO WEST. "I can't go in that direction". WEST. "I can't go in that direction".

Novice reads the text again, which clearly states there is a door to the west. He is just about to learn Rule 3. Frequently games abbreviate movement to the compass points, usually N,S,E and W, although occasionally also the NW, NE, SE and SW come into play.

This does not necessarily mean that the game will recognise the full word. W, types Novice, and off he goes into another room, with exits North and East.

I think by now that you get the general idea, so that we can leave Novice to stumble around and explore by himself for a while.

The whole point is that to progress in adventures you need to try and understand exactly what the computer is doing to analyse your input.

Once you understand that then you begin to see how you can avoid spending half your adventure reading the same boring "I don't understand" messages.

What the computer does is store each chunk of your input and compare it with a list of words that it is programmed to understand.

In a simple verb/noun input analyser, the first chunk will be compared to a list of verbs, and if the computer has that verb the program will branch off to check the noun.

In the example above, the computer checked through a list of verbs and found READ. After that it repeated the process with BOOK in the list of nouns. Simple, huh?

So now you can see why Novice had this first problem. The computer analysed his input as READ THE. A quick check through the list reveals no such noun as THE, and the computer sends its message.

In some games such as The Pay-Off, the message will tell you where you are going wrong – "You can't READ a THE".

However in many adventures the same error message is repeated time and time again with no clue as to the problem.

It gets more complicated than that though. In a bid to save memory, and therefore add more to the game, the input analyser will tend to only recognise the first three or four letters of each word.

This means that GET BRONTOSAURUS is exactly the same as GET BRON. So you don't always need to type out great long words to be understood.

And now we start to see how these new complicated analysers work. What they actually do is to ignore most of what you type, noting your action, what you are doing it to, and whether it involves some form of modifier, such as LOOK BEHIND THE DESK.

To my mind therefore, the use of full sentence analysis merely allows the player to type in a more normal, real life instruction, with little or no effect on the likely response, but more opportunity for a typing error.

It does allow the player to input a

string of commands, to be actioned in sequence, which allows a player to move rapidly through sections of the game already completed and that must be a help.

But aside from that, the temptation to type GET BOOK rather than PICK UP THE DUSTY RED BOOK FROM THE TOP SHELF, gets me every time.

Now I note from the letters page that at least one person is having a bit of trouble with Lords of Time from our old friends Level 9.

No surprises there, as this is another huge game set across nine time zones, with a vital artefact to collect from each zone.

Without further details of where you get stuck I'm afraid Brillig's help can only be guesswork, so if you do have a problem please give me an idea of what you have already covered.

A map would be of assistance as well. And if you have any completed adventure maps send them in – you could help save someone's sanity.

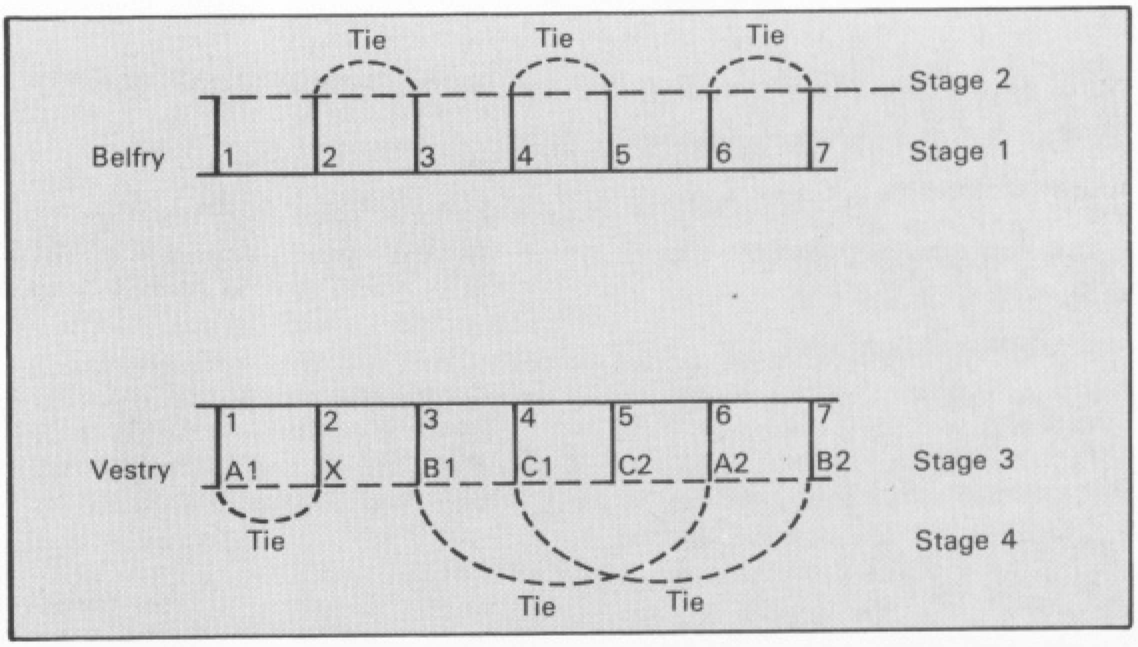
Just in case the problem in Lords of Time is getting started, don't forget that a clock sometimes stops and needs a helping hand. Also Level 9 use a slightly different interpretation of IN than most of us. It won't tell you what you are carrying, but may move you in mysterious ways.

Speaking of Level 9, I mentioned last month that the interpreter they use gives out the occasional glitch. So I have decided to begin a "Glitch of the Month" competition, with an Atari User T shirt as the prize.

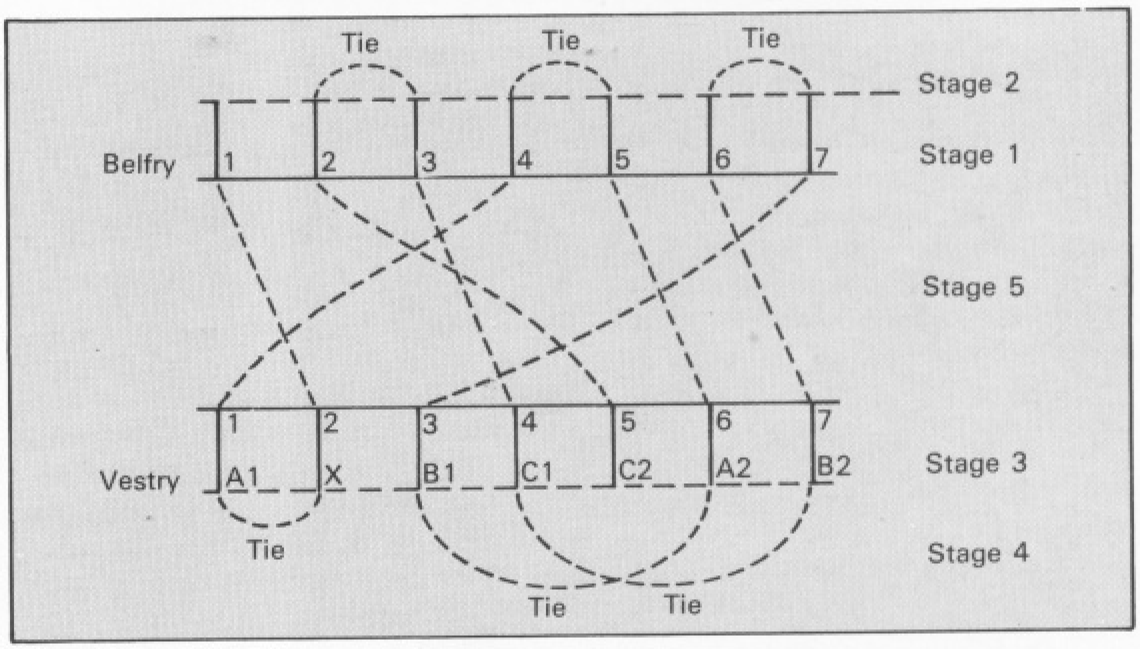
Send your favourite glitch to Atari User, together with your size, and every month I'll announce a winner.

This month, as the contest has only just started, the winner is me, and as we have just looked briefly at Lords of Time, try typing in "Get All" and then try to puzzle out what you would need an Allosaurus for. Sorry boys it may seem like victimisation but anyone who uses "Arfle barfle gloop" as an error message is really asking for all they get.

One last piece of news. Scott Adams' Questprobe 3 . . . The Fantastic Four was due out in July. I'll take a serious look at the series to date next month.



Unravelling that tangle in the belfry



IN last month's problem we left you with some bell ropes to sort out after Quasimodo had left them in a tangle.

The solution is as follows – and you would be well advised to make a map:

1. In the belfry, number the ropes 1-7.
2. Tie 2-3, 4-5, and 6-7.
3. Go down to the vestry, number the ropes 1-7, then pull any rope. If another rope goes up mark the pairs A1-A2, B1-B2 and so on. If nothing happens, that rope must be number 1, identify it with an X.
4. Tie X-A1, A2-B1, and B2-C1.
5. Return to the belfry and pull rope 1. The rope that moves down is A1 so you can mark that and A2 (you tied them together in step 2).
6. Pull A2 to identify B1 and B2 and so on.
7. Go down, at which point the program says "Quasimodo ties the 7 ropes to the 7 bells".
8. Pull the ropes in the order that will ring them 1-7. Our particular solution was 2,5,4,1,6,7,3 but yours could be different due to the random element.

Drop a brick and solve a problem

AN essential element, indeed almost a definition, of an adventure is that the player moves around, picking up and dropping objects until the game is solved and the program stops.

If you think you have a few good ideas but don't know how to start, you could try writing a program to solve the following problem.

You are as usual, alone in a system of caves/rooms that stretch into the distance to the East and the West. You are carrying a hod of bricks and five cards on which is written the following:

Card	Empty	Not empty
1	DROP,E,2	GET,W,3
2	DROP,E,3	GET,E,5
3	DROP,W,1	GET,E,2
4	DROP,W,3	E,1
5	E,4	STOP

All the caves are empty and all you have to do is take the role of the computer for a change – in other words obey the instruction on the cards beginning with card 1 which says:

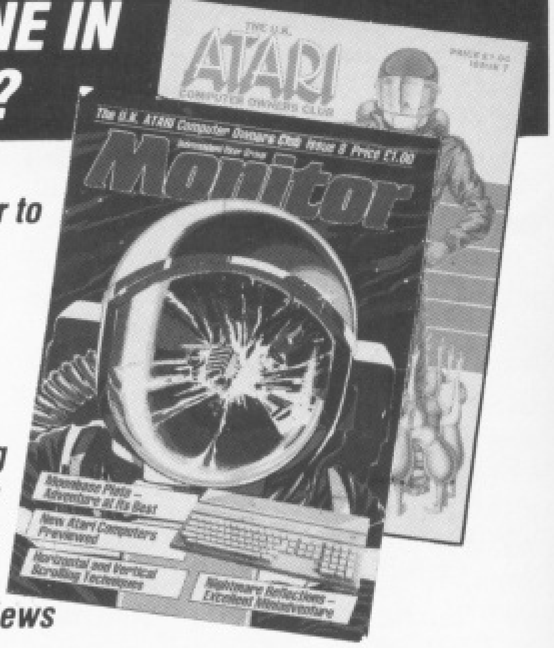
IF this cave is Empty THEN Drop a brick, go East, and obey card 2, alternatively Get the brick, go West and obey card 3.

If you follow the instructions correctly you will eventually reach the STOP on card 5. The questions are:
a) How many bricks will you drop, and
b) How many E/W moves will you make?

A couple of hints – you should never have to go further than 10 caves to the West. Nevertheless, don't try solving the problem with real bricks, there are a lot of moves involved and you probably will need to write a program to solve it.







Try life in the RAW and you'll opt out of op codes

ROLAND WADDILOVE takes some of the toil out of machine code programming with his RAW assembler

AS you probably know, a machine code program consists of a series of binary numbers in the range 0 to 255, although we normally use hexadecimal or decimal as it's easier to follow.

Even so, programs are very difficult to read. For example, what does $A5 $D4 mean? Very little I should imagine, unless you know all the opcodes off by heart.

Assembly language is much easier to digest. A mnemonic is used to represent each machine code instruction.

For example, the codes above can be represented by:

LDA $D4

which is much more meaningful. It's not perfect but it's a big improvement.

.byte=$D4
LDA byte

is even better.

What an assembler does is to convert these assembly language mnemonics into machine code for you, taking all the hard work and tedium out of it. There's no need to look up the opcodes at all.

Assembly listings are easier to follow and much easier to debug if they don't work first time – and they rarely do.

RAW, the assembler presented here, will allow you to write assembly language programs. The assembly listing can be saved along with RAW and the machine code run using the USR function.

Listing I shows an example of what is possible with RAW and demonstrates some of its functions. It's a short program to convert any upper case letters in a string to lower case.

To use it, enter:

X=USR(1616,ADR(A$), LEN(A$))

where A$ is the string to be converted after assembling the routine.

The assembly listing is entered as a series of data statements. Multiple statement lines are possible by separating the statements with commas and comments can be included by placing them in REM statements.

The first part of an assembler instruction is always three letters. This must be followed by one space if there is a further part. The comma in indexed instructions should be replaced by a full stop otherwise RAW will get confused.

Implied instructions are always one part. For example:

RTS

All other instructions are two part. Like this:

ASL A
LDA 27

```
10 DATA ORG $650
20 DATA .t1=$D4
30 DATA .t2=$D5
40 DATA PLA
50 DATA PLA
60 DATA STA t2
70 DATA PLA
80 DATA STA t1
90 DATA PLA
100 DATA PLA
110 DATA TAY
120 DATA DEY
130 DATA .loop
140 DATA LDA (t1).Y
150 DATA CMP #65
160 DATA BCC next
170 DATA CMP #91
180 DATA BCS next
190 DATA ORA #$60
200 DATA STA (t1).Y
210 DATA .next
220 DATA DEY
230 DATA BPL loop
240 DATA RTS
250 DATA END
260 REM
```

Listing I

The rules are strict but you'll find RAW friendly

Notice the single space between the first and second part. Indexed instructions are also two part, such as:

LDA ($D4),Y
CMP byte,X

The commas need to be replaced by full stops for RAW to understand them, so they become:

LDA ($D4).Y
CMP byte.X

Numbers can be either decimal or hexadecimal. Hex numbers are preceded by $. A label can be used instead of a number and it's possible to have a forward reference to a label as it's a two pass assembler.

The first instruction must be ORG followed by a number. This tells RAW where to place the object code.

The first 55 bytes of Page 6 in the memory are used by the assembler but the rest is free. So:

ORG $650

will set the object code address to 1616 decimal, $650 hex.

Space for the object code can be reserved by moving RAMTOP down. RAMTOP is the highest point in memory available to Basic, and

? PEEK(106)*256

will tell you what it is currently set to. To reserve 1k of memory – four pages – use:

POKE(106),PEEK(106)–4

then enter a GRAPHICS command to relocate the display list and data.

Please note however, that the first 800 bytes above RAMTOP may be corrupted by scrolling a text window, using CLEAR or clearing the screen

```
5000 REM 6502 Assembler
5010 REM By R.A.Waddilove
5015 REM (c) Atari User
5020 GOSUB 9010
5025 FOR PASS=0 TO 1
5030 ? CHR$(125):? "RAW 6502 Assembler
.....Pass ";PASS+1:?
5040 RESTORE :READ A$
5070 IF A$(1,3)<>"ORG" THEN ERR=3:GOTO
 10000
5080 IF A$(1,3)="ORG" THEN T$=A$(5):GO
SUB 6010:P=X:START=P
5090 IF PASS THEN POKE 85,20:? A$:X=IN
T(P/256):GOSUB 6500:X=P-256*X:GOSUB 65
00:? ":";
5100 N=0:READ A$
5110 IF A$ ="END" THEN 5400
5120 IF A$(1,1)="." THEN GOSUB 8210:GO
TO 5090
5130 IF A$(1,3)="DEF" THEN GOSUB 8500:
GOTO 5090
5200 GOSUB 6600:GOSUB 7150:A$(4,4)=" "
5210 IF BYTE>-1 THEN POKE P,BYTE
5220 IF BYTE1>-1 THEN POKE P+1,BYTE1
5230 IF BYTE2>-1 THEN POKE P+2,BYTE2
5250 P=P+N:GOTO 5090
5400 NEXT PASS
5410 POKE 85,20:? A$:? :? "Errors=0  S
tart=";START;"  Length=";P-START+1
5500 END
6000 REM ---- Get Value ---------
6010 IF T$(1,1)>"$" THEN 6100
6020 X=0:J=0
6030 FOR I=LEN(T$) TO 2 STEP -1
6040 IF T$(I,I)>"9" THEN X=X+(ASC(T$(I
,I))-55)*Z(J)
6050 IF T$(I,I)<"A" THEN X=X+(ASC(T$(I
,I))-48)*Z(J)
6060 J=J+1
6070 NEXT I
6080 RETURN
6100 IF T$<"A" THEN X=VAL(T$):RETURN
6120 IF LEN(T$)<4 THEN T$(LEN(T$)+1)="
   "
6130 X=USR(1536,ADR(T$),ADR(L$))
6140 IF X<255 THEN X=V(X):RETURN
6150 IF PASS THEN ERR=4:GOTO 10000
6160 X=P
6200 RETURN
6400 REM ----- Hex print X ------
6500 IF  NOT PASS THEN RETURN
6510 I=1+X-16*INT(X/16):J=1+INT(X/16):
? H$(J,J);H$(I,I);:RETURN
6590 REM ----- Addressing mode ---
6600 IF LEN(A$)=3 THEN A$(LEN(A$)+1)="
I":N=1:RETURN
6620 IF A$(5,5)="A" THEN A$(4,4)="A":N
=1:RETURN
6630 IF A$(5,5)="#" THEN A$(4,4)="B":N
=2:T$=A$(6):RETURN
6640 IF A$(LEN(A$)-1)="X)" THEN A$(4,4
)="K":T$=A$(6,LEN(A$)-3):N=2:RETURN
6650 IF A$(LEN(A$)-2)=").Y" THEN A$(4,
4)="L":T$=A$(6,LEN(A$)-3):N=2:RETURN
6660 IF A$(LEN(A$))=")" THEN A$(4,4)="
M":T$=A$(6,LEN(A$)-1):N=3:RETURN
6670 IF A$(1,1)="B" AND A$(2,2)<>"I" T
HEN A$(4,4)="J":T$=A$(5):N=2:RETURN
6680 IF A$(LEN(A$))<>"X" THEN 6700
6682 T$=A$(5,LEN(A$)-2):GOSUB 6010
6684 IF X<256 THEN A$(4,4)="D":N=2:RET
URN
6686 A$(4,4)="G":N=3:RETURN
6700 IF A$(LEN(A$))<>"Y" THEN 6720
6702 T$=A$(5,LEN(A$)-2):GOSUB 6010
6704 IF X<256 THEN A$(4,4)="E":N=2:RET
URN
6706 A$(4,4)="H":N=3:RETURN
6720 T$=A$(5):GOSUB 6010
6724 IF X<256 THEN A$(4,4)="C":N=2:RET
URN
6730 A$(4,4)="F":N=3:RETURN
7000 REM --- Search -----
7150 BYTE1=-1:BYTE2=-1
7190 BYTE=USR(1536,ADR(A$),ADR(M$))
7290 X=BYTE:GOSUB 6500
7305 IF X=255 THEN ERR=1:GOTO 10000
7310 IF N=1 THEN RETURN
7320 GOSUB 6010:? " ";
7330 IF N=3 THEN BYTE2=INT(X/256):BYTE
1=X-256*BYTE2:X=BYTE1:GOSUB 6500:? " "
;:X=BYTE2:GOSUB 6500:RETURN
7340 IF A$(4,4)<>"J" THEN BYTE1=X:GOSU
B 6500:RETURN
7350 IF X>255 THEN X=X-P-2
7360 IF X>255 THEN ERR=2:GOTO 10000
7370 IF X<0 THEN X=X+256
7380 BYTE1=X:GOSUB 6500
7500 RETURN
```

with:

? CHR$(125)

The last assembler instruction must be END. This tells RAW to stop assembling!

Labels are defined by preceding them with a full stop. They can be up to four characters long and must be lower case letters. Numbers can be included in the name. Up to 255 labels can be defined.

A label can be set to the current object code address or to any positive value, so:

.loop

will set *loop* to the current address. This can be done at any point in the assembler listing.

.num=123

will set *num* to 123. If labels are to be set to particular values in this way it must be done at the start of the program.

Bytes, words and strings can be placed in the memory at the current object code address. Strings must be enclosed by single quotes.

DEFB $40
DEFW 16384
DEF$ 'Assembler'

The program has been numbered starting at line 5000. This is to allow 1 to 4999 to be used for the assembly listing in DATA statements.

It's not very long and the program uses subroutines to find the addressing mode, labels, opcodes and so on. Each subroutine has a title describing its function.

There's a short machine code subroutine which is simply a super fast string search. It's used to find mnemonics and labels.

Please note the rules – they're quite strict. Apart from that you'll find RAW to be friendly, functional and extremely flexible.

Happy assembling!

VARIABLES

P	Object code address.
BYTE **BYTE1** **BYTE2**	Object code.
A$	Assembler instruction.
M$	Mnemonics.
H$	Hex digits.
L$	Label names.
V( )	Values of the labels.
T$	Label or number in A$

```
8200 REM --- Define label -------
8210 IF PASS THEN RETURN
8220 J=0
8230 FOR I=2 TO LEN(A$)
8240 IF A$(I,I)="=" THEN J=I
8250 NEXT I
8260 IF J=0 THEN L$(L*4+1,L*4+4)=A$(2,
LEN(A$)):V(L)=P:L=L+1:RETURN
8270 L$(L*4+1,L*4+4)=A$(2,J-1):T$=A$(J
+1):GOSUB 6000:V(L)=X:L=L+1
8400 RETURN
8500 REM --------- DEF ----------
8510 IF A$(4,4)="$" THEN 8600
8520 T$=A$(6):GOSUB 6000
8530 IF A$(4,4)="B" THEN GOSUB 6500:PO
KE P,X:P=P+1:RETURN
8540 IF A$(4,4)="S" THEN P=P+X:RETURN
8550 BYTE=X-256*INT(X/256):BYTE1=INT(X
/256):X=BYTE:GOSUB 6500:? " ";:X=BYTE1
:GOSUB 6500
8560 POKE P,BYTE:POKE P+1,BYTE1:P=P+2:
RETURN
8600 FOR I=6 TO LEN(A$)
8610 POKE P,ASC(A$(I,I)):P=P+1
8620 NEXT I
8630 RETURN
9000 REM ----- Mnemonics --------
9010 DIM A$(15),T$(5),H$(16),L$(1024),
V(255),Z(3)
9015 Z(0)=1:Z(1)=16:Z(2)=256:Z(3)=4096
9020 H$="0123456789ABCDEF"
9030 L$=" ":L$(1024)=" ":L$(2)=L$:L=0
9080 RESTORE 9090:FOR I=0 TO 54:READ J
:POKE 1536+I,J:NEXT I
9090 DATA 104,104,133,213,104,133,212,
104,133,215,104,133,214,162,0,160
9091 DATA 3,177,214,209,212,208,10,136
,16,247,134,212,169,0,133
9092 DATA 213,96,24,165,214,105,4,133,
214,165,215,105,0,133,215,232,208
9093 DATA 222,134,213,202,134,212,96
9100 DIM M$(1024)
9110 M$="BRKIORAK............ORACASLC.
...PHPIORABASLA........ORAFASLF...."
9120 M$(LEN(M$)+1)="BPLJORAL..........
..ORADASLD....CLCIORAH............ORAG
ASLG...."
9130 M$(LEN(M$)+1)="JSRFANDK........BI
TCANDCROLC....PLPIANDBROLA....BITFANDF
ROLF...."
9140 M$(LEN(M$)+1)="BMIJANDL..........
..ANDDROLD....SECIANDH............ANDG
ROLG...."
9150 M$(LEN(M$)+1)="RTIIEORK..........
..EORCLSRC....PHAIEORBLSRA....JMPFEORF
LSRF...."
9160 M$(LEN(M$)+1)="BVCJEORL..........
..EORDLSRD....CLIIEORH............EORG
LSRG...."
9170 M$(LEN(M$)+1)="RTSIADCK..........
..ADCCRORC....PLAIADCBRORA....JMPHADCF
RORF...."
9180 M$(LEN(M$)+1)="BVSJADCL..........
..ADCDRORD....SEIIADCH............ADCG
RORG...."
9190 M$(LEN(M$)+1)="....STAK........ST
YCSTACSTXC....DEYI....TXAI....STYFSTAF
STXF...."
9200 M$(LEN(M$)+1)="BCCJSTAL........ST
YDSTADSTXE....TYAISTAHTXSI........STAG
........"
9210 M$(LEN(M$)+1)="LDYBLDAKLDXB....LD
YCLDACLDXC....TAYILDABTAXI....LDYFLDAF
LDXF...."
9220 M$(LEN(M$)+1)="BCSJLDAL........LD
YDLDADLDXE....CLVILDAHTSXI....LDYGLDAG
LDXH...."
9230 M$(LEN(M$)+1)="CPYBCMPK........CP
YCCMPCDECC....INYICMPBDEXI....CPYFCMPF
DECF...."
9240 M$(LEN(M$)+1)="BNEJCMPL..........
..CMPDDECD....CLDICMPH............CMPG
DECG...."
9250 M$(LEN(M$)+1)="CPXBSBCK........CP
XCSBCCINCC....INXISBCBNOPI....CPXFSBCF
INCF...."
9260 M$(LEN(M$)+1)="BEQJSBCL..........
..SBCDINCD....SEDISBCH............SBCG
INCG...."
9500 RETURN
10000 REM ----- Errors -----
10010 A$(4,4)=" ":? :?
10020 IF ERR=1 THEN ? "Couldn't find "
;
10030 IF ERR=2 THEN ? "Jump too far ";
10040 IF ERR=3 THEN A$="Origin?"
10050 IF ERR=4 THEN ? T$;" Not defined
 ";
10090 ? A$:END
```

Part II of MIKE ROWE's series on how to give your program displays the professional touch

There are ways round those dirty programming techniques

A CUSTOM display list, mixing several modes on the same screen, can quickly and easily give your display a professional touch.

There are two ways to create one. Firstly you can modify a standard display list created by the operating system after a Graphics call. Secondly you can create an entirely new list from scratch, or even have several display lists in memory at the same time.

Before you start to construct your list there are several problems to be considered.

If you are modifying an existing display list it is safest to use the graphics mode that takes up most memory in your final display list as a starting point for your modified list.

Also try to avoid your screen memory crossing a 4k boundary – 4k, 8k, 12k, and so on to 48k – as it will cause problems. If you must cross a border, say if an 8k mode is used, then when the screen reaches the boundary you need to insert another load memory scan – see last month's article – in the display list to point to the start of the next 4k block of screen RAM.

Different graphics modes take up a different amount of screen RAM per line. If the operating system expects a line to take 40 bytes and in the modified list a line takes only 20, then the data below this line will be shifted half way across the screen.

There are two ways of avoiding problems with this. First you can use "dirty programming" and design your new lines in groups of lines which add to make the correct number of bytes – see examples later.

The other way is to avoid using the operating system for Drawto, Plot or Print commands and poke directly to screen memory.

If you are to use Basic commands such as Plot, Drawto or Print on the screen you may need to fool the OS into thinking it is drawing on the correct screen.

This is done by poking location 87 ($57) with the Basic graphics mode of the line involved.

Second is the problem of Basic checking each command to check that it is in the range allowed by the graphics mode it thinks is in use.

This can commonly lead to Basic thinking it is going to print off the screen and giving an error when you know full well that it is on the screen.

This is solved by tampering with locations 88 and 89. These contain the location of the top left corner of screen memory and the OS uses these to calculate the legality of a screen command.

The top corner can be calculated by PEEK(88)+PEEK(89)*256. If these locations are poked with the memory location of the start of the line to which you want to plot or print, then the start of this line becomes position 0,0 and therefore within legal range.

Knowing the number of bytes taken up per line in each mode is therefore needed as it is for the second point above.

Basic mode	Antic mode	Bytes per line
0	2	40
1	6	20
2	7	20
3	8	10
4	9	10
5	10	20
6	11	20
7	13	40
8	15	40
9	15	40
10	15	40
11	15	40

On to some examples. The

```
0 REM CUSTOM DISPLAY LIST DEMO 1
10 GRAPHICS 0:REM GET OS TO CREATE DIS
PLAY LIST
20 DLIST=PEEK(560)+PEEK(561)*256:REM F
IND START OF DISPLAY LIST
30 POKE DLIST+3,71:REM CHANGE 1ST GRAP
HICS INSTRUCTION FROM 66 (64+2=LMS+GRA
PHICS 0) TO 64+7 (GRAPHICS 2)
40 POKE DLIST+6,6:REM CHANGE LINE 2 FO
RM 2 (GRAPHICS 0) TO 6 (GRAPHICS 1)
50 REM CHANGE 2 LINES TO AVOID SHIFTIN
G THE REST OF THE DISPLAY BY HALF A SC
REEN
100 POSITION 3,0:? "  GRAPHICS 0
     title "
```

Program I

simplest way to write a modified list is shown in Program I. This will add two lines for a larger, coloured title to the top of a Graphics 0 screen.

It works, but again it is dirty programming. The maximum number of scan lines allowed in a display list is usually 192. This display list is more than 192 scan lines long.

In reality Antic can cope with slightly more lines than the theoretical maximum. I have found that an extra 24 usually is stable, but more than this and the screen will roll.

See last month's article for a table of the number of scan lines for each mode line.

A better programming technique would be to calculate the number of scan lines being used and make sure that the total is 192 or less. This will usually involve moving the end of the display list and rewriting it as in Program II.

As can be seen, the end of the display list is indicated by a number 65 – $41. The two numbers following this are the location of the start of the display list in the order Low Byte, High Byte. Therefore the first number can be found by PEEK(560) and the second by PEEK(561), as these should be the same.

The third way is to create your own list from scratch. This is how virtually all machine code programs get their displays and one of the reasons that they can be so spectacular.

If you avoid using the OS to draw to the screen then many of the limitations of custom display lists also

```
0 REM CUSTOM DISPLAY LIST DEMO 2
10 GRAPHICS 7:REM MOST MEMORY GREEDY M
ODE USED
20 DLIST=PEEK(560)+PEEK(561)*256:REM F
IND START OF DISPLAY LIST
30 LOW=PEEK(88):HIGH=PEEK(89):REM MEMO
RY LOCATIONS OF START OF SCREEN
40 SETCOLOR 3,3,10
100 POKE DLIST+3,64+7:REM COVERT 1st L
INE TO GRAPHICS 2
110 FOR I=DLIST+6 TO DLIST+53:POKE I,1
3:NEXT I:REM NEXT 48 LINES GRAPHICS 7
(ACTUALLY ALREADY 13 - COULD LEAVE)
120 POKE DLIST+54,6:POKE DLIST+55,6:RE
M 2 LINES OF GRAPHICS 1 (N.B 2 LINES K
EEPS LINE MEMORY REQUIREMENTS AS 40)
130 POKE DLIST+56,4:REM 1 LINE OF GRAP
HICS 12(WILL WORK ON NON-XLS)
140 FOR I=DLIST+57 TO DLIST+63:POKE I,
2:NEXT I:REM 7 LINES OF GRAPHICS 0
150 POKE DLIST+64,65:REM END OF DISPLA
Y LIST
160 POKE DLIST+65,PEEK(560):POKE DLIST
+66,PEEK(561):REM POINT BACK TO START
OF DISPLAY LIST
200 POSITION 0,0:POKE 87,2:? #6;" DISP
LAY LIST DEMO":REM PRINT ON TOP LINE
210 LOW=LOW+20:IF LOW>255 THEN LOW=LOW
-256:HIGH=HIGH+1:REM RECALCULATE NEW T
OP LEFT CORNER 1 LINE DOWN
220 POKE 88,LOW:POKE 89,HIGH:POKE 87,7
:REM TELL OS THAT 2ND LINE IS TOP OF S
CREEN AND TELL OS THAT IN MODE 7
230 COLOR 2:PLOT 0,24:FOR I=0 TO 159:D
RAWTO I,SIN(I*9)*20+24:NEXT I:REM DRAW
 DESIGN
235 COLOR 3:PLOT 0,0:DRAWTO 159,0:DRAW
TO 159,47:DRAWTO 0,47:DRAWTO 0,0
240 LOW=LOW+48*40
250 IF LOW>255 THEN LOW=LOW-256:HIGH=H
IGH+1:GOTO 250
260 POKE 88,LOW:POKE 89,HIGH:REM RECAL
CULATED NEXT LINE MEMORY LOCATION AND
FOOL OS THAT IT'S THE TOP OF SCREEN
270 POKE 87,1:REM TELL OS THAT IT'S GR
APHICS 1
280 POSITION 0,0
290 ? #6;"     by mike rowe":REM INVERS
E
300 ? #6;"        1985"
310 POKE 87,1:FOR I=1 TO 40:? #6;CHR$(
42);:NEXT I
315 POKE 752,1
320 POKE 87,1:? #6;" This demonstratio
n shows a simple kind  of multiple mod
e display that can be "
330 ? #6;" obtained by customising an
existing     display list."
```

Program II

```
0 REM PROGRAM 3
1 REM MIKE ROWE 1985
2 REM ENTER PROGRAM EITHER WITH ALL RE
MS OR MISS ALL OF THEM OUT!
10 DIM DLI$(50),SCREEN$(1000)
20 REM STORE DISPLAY LIST AND SCREEN M
EMORY IN STRINGS
100 SC=ADR(SCREEN$):SCHIGH=INT(SC/256)
:SCLOW=SC-SCHIGH*256:REM CALCULATE LOW
 AND HIGH BYTES OF SCREEN MEMORY
110 DL=ADR(DLI$):DLHIGH=INT(DL/256):DL
LOW=DL-DLHIGH*256:REM AND AGAIN FOR AD
DRESS OF DISPLAY LIST
120 POKE 560,DLLOW:POKE 561,DLHIGH:REM
 TELL ANTIC THE START OF THE DISPLAY L
IST
200 FOR I=1 TO 25:READ A:DLI$(I)=CHR$(
A):NEXT I:REM READ IN DATA FOR DISPLAY
 LIST
210 POKE DL+4,SCLOW:POKE DL+5,SCHIGH:R
EM TELL DISPLAY LIST START OF SCREEN M
EMORY
220 POKE DL+24,DLLOW:POKE DL+25,DLHIGH
:REM POINT END OF DISPLAY LIST BACK TO
 START
300 SCREEN$(1)=CHR$(0):SCREEN$(1000)=C
HR$(0):SCREEN$(2)=SCREEN$:REM ZEROS IN
 ALL OF SCREEN MEMORY
310 SCREEN$="@@@custom@display@@@@@@@@
@list@demo":REM INVERSE IN QUOTES
320 FOR I=40 TO 59:POKE I+SC,72:NEXT I
321 FOR I=60 TO 79:POKE I+SC,33:NEXT I
322 FOR I=80 TO 99:POKE I+SC,72:NEXT I
323 FOR I=100 TO 119:POKE I+SC,33:NEXT
 I
324 FOR I=120 TO 139:POKE I+SC,72:NEXT
 I
325 FOR I=140 TO 159:POKE I+SC,33:NEXT
 I
326 FOR I=160 TO 179:POKE I+SC,72:NEXT
 I
327 FOR I=180 TO 199:POKE I+SC,33:NEXT
 I
328 FOR I=200 TO 219:POKE I+SC,72:NEXT
 I
330 SCREEN$(259)=CHR$(16)
340 SCREEN$(290)="Q"
350 SCREEN$(310)="R":REM INVERSE R
360 REM ALL ABOVE IS THE DATA BEING TO
 BE HELD IN SCREEN$ TO GIVE THE SIMPLE
 DISPLAY
1000 DATA 112,112,112,71,0,0,7,112,10,
10,10,10,10,10,10,10,10,10,112,2,6,7,6
5,0,0
```

Program III

disappear. However the other side of the coin is that the OS no longer does the hard work for you and the programming becomes more difficult.

Program III demonstrates both these points but to keep it short does not do justice to the capabilities of your Atari.

As I mentioned previously, Graphics modes 12-15 are only directly available on the XL and XE Ataris. However all the machines are capable of displaying these modes.

Many commercial games in fact use Graphics 12, Antic mode 4. The two most useful of these modes, 12 and 15, can be obtained using programs IV and V.

Program VI is just a little bit of lunacy for light relief.

This is a brief overview of custom display lists and gives some idea of how they can improve the appearance of a simple screen.

However to bring it to life you can use Display List Interupts to achieve numerous special effects. We will discuss this next month.

```
10 REM CUSTOM DISPLAY LIST DEMO 5
1000 REM PROGRAM TO CONVERT GRAPHICS 8
 TO GRAPHICS 15 (ANTIC 14) FOR THE ATA
RI 400/800
1010 GRAPHICS 8:REM START WITH GRAPHIC
S 8 (OR GRAPHICS 8+16) DISPLAY LIST
1020 DLIST=PEEK(560)+PEEK(561)*256
1030 POKE DLIST+3,78
1040 FOR I=6+DLIST TO 198+DLIST:REM RE
ST OF DISPLAY LIST
1050 IF PEEK(I)=15 THEN POKE I,14:REM
CONVERT NORMAL GRAPHICS 8 LINE TO GRAP
HICS 15
1060 IF PEEK(I)=79 THEN POKE I,78:REM
DISPLAY LIST CONTAINS SOME LMS COMMAND
S AS SCREEN OVER 4K SO CONVERT THESE
1070 NEXT I
1080 POKE 87,7:REM FOOL OS INTO THINKI
NG THAT IT IS IN GRAPHICS 7
1081 REM HOWEVER OS WILL ONLY LET YOU
PLOT IN THE UPPER HALF OF THE SCREEN
1082 COLOR 1:PLOT 0,0:DRAWTO 159,95
1090 REM ENTER NEXT LINE TO PLOT TO LO
WER HALF OF SCREEN
1091 POKE 89,PEEK(89)+15:REM FOOLS OS
INTO THINKING SCREEN STARTS HALF WAY D
OWN ACTUAL SCREEN
1092 COLOR 1:PLOT 0,0:DRAWTO 159,95
```

Program V

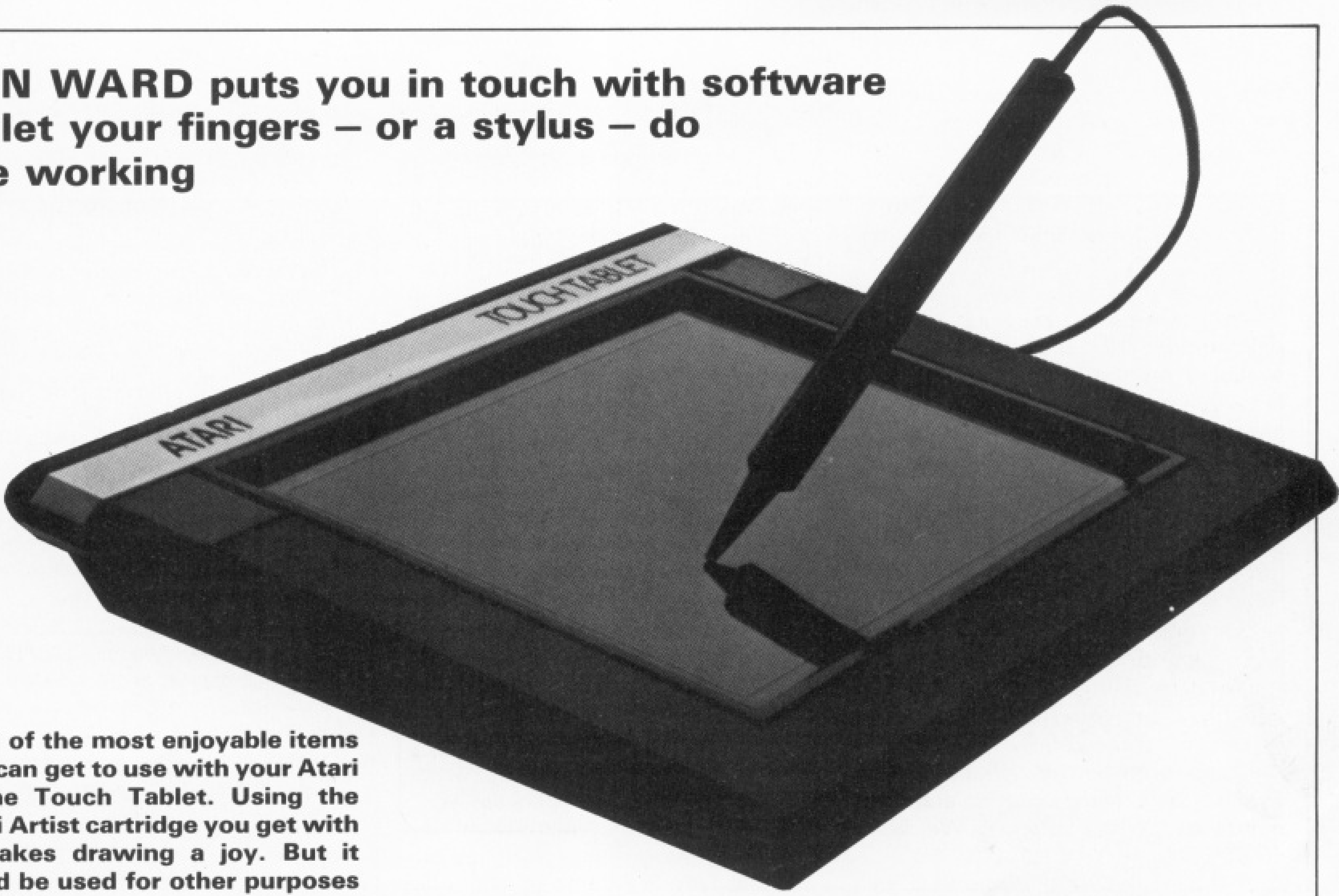
```
10 REM CUSTOM DISPLAY LIST DEMO 4
20 REM TO CREATE GRAPHICS 12 (ANTIC MO
DE 4) EVEN ON ATARI 400/800s
30 REM ACTS LIKE MULTICOLOUR GRAPHICS
0 - ON XL'S ACTS LIKE GRAPHICS 1 WITH
OR WITHOUT A WINDOW
40 REM IF THIS IS REQUIRED START WITH
A GRAPHICS 5 DISPLAY LIST AND CONVERT
THIS
1000 GRAPHICS 0:REM START WITH GRAPHIC
S 0 DISPLAY LIST
1010 DLIST=PEEK(560)+PEEK(561)*256:REM
 START OF DISPLAY LIST
1020 POKE DLIST+3,68:REM CONVERT 1ST L
INE TO ANTIC 4
1030 FOR I=DLIST+6 TO DLIST+28:POKE I,
4:NEXT I:REM CONVERT REST OF LIST TO A
NTIC 4
```

Program IV

```
0 REM CUSTOM DISPLAY LIST DEMO 6
10 FOR I=1 TO 100
20 POKE 561,PEEK(53770)
30 FOR WAIT=1 TO 20:NEXT WAIT
40 NEXT I
100 GRAPHICS 0
110 ? :? "GOT YOU WORRIED!!"
```

Program VI

KEN WARD puts you in touch with software to let your fingers – or a stylus – do the working



Take a tablet for more than drawing

ONE of the most enjoyable items you can get to use with your Atari is the Touch Tablet. Using the Atari Artist cartridge you get with it makes drawing a joy. But it could be used for other purposes as well.

The Touch Tablet can be used anywhere a joystick, paddle, light pen or mouse would be used, and it's faster than most.

You don't have to drag the cursor across the screen as with the joystick, paddle or mouse. You place your pen where you want it straight away.

And the advantage over the light pen is that you don't have to pick anything up – you just use your finger or the stylus.

The only problem is that at the moment not a lot of software is available for it. The only commercial program I know of that has a Touch Tablet option is The Music Construction Set from Electronic Arts.

Having tried it, I can say that the program certainly is a lot easier to use with the Touch Tablet than with a joystick. So it's up to us to supply our own programs until the software houses get around to it.

OK, so where do we start? The tablet can be read by:

PADDLE(0): *Horizontal readings*
PADDLE(1): *Vertical readings*

The readings given are between 1 and 228. The horizontal ones are from left, 1, to right, 228, and the vertical ones from bottom 1, to top 228. So we have two problems to overcome before using these readings.

- They have to be related to screen positions – we don't have a 228 by 228 pixel screen to work with. Also in converting the figures to relate to the screen we have to invert the vertical readings. If we are going to use a player/missile for the cursor, we have the added problem of coordinating P/M positions and screen position.
- We have to deal with the cursor wobble common to all variable resistance input devices.

The triggers on the tablet can be read at PTRIG(0) and 1 or at STICK(0). The trigger on the plug-in pen can only be read at STICK(0). So for simple inputs it is easier to use STICK(0) . If STICK(0)<15 then one of the triggers has been pressed.

For example, let's assume we are going to work in Graphics 0. The first thing to do is to relate the 228 reading from the tablet to the 40 characters on the screen.

We could simply divide 228 by 40 which gives us a divisor of 5.7, which would work fine apart from one small problem – it means that for the extreme left and right positions we would be right up against the frame of the tablet.

The ideal area to work in is marked on the tablet, which is far enough in to allow even the biggest finger to get to the edge. Remember, not everyone will want to use the stylus.

So let's look at it in practice:

```
10 GRAPHICS 0
20 X=INT(PADDLE(0)/5.3)-3
30 Y=24-INT(PADDLE(1)/8.87)
40 POSITION 10,10:? X,Y;" "
50 GOTO 20
```

If you try this out you'll see that we now have an additional problem –

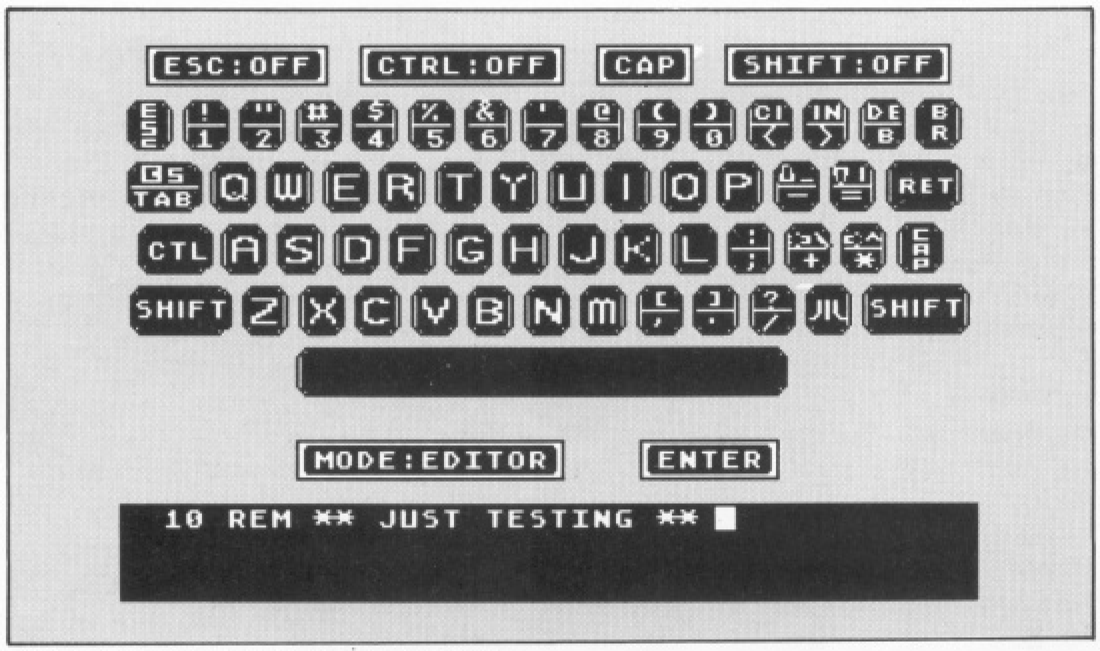


Figure I: Using the touch tablet as a keyboard

readings outside the screen range. But it's only a minor one. We can overcome it with a series of IF X<0 . . . IF X>39 . . . IF Y<0 . . . IF Y>39 . . .

If that was all there was to it we would all be churning out Touch Tablet programs.

Try this one and you'll see the other problems:

```
10 GRAPHICS 7
20 X=INT(PADDLE(0)/1.3)-11
30 Y=INT((215-PADDLE(1))/2.1)
40 IF X<0 THEN X=0
50 IF X>159 THEN X=159
60 IF Y<0 THEN Y=0
70 IF Y>79 THEN Y=79
80 COLOR 1:PLOT X,Y
90 GOTO 20
```

The first is that you have to go slowly to draw a continuous line, and if you were flashing a cursor and checking for trigger and/or key presses it would be even slower.

The second problem is the odd random pixel being drawn as you lift and lower the stylus to the pad.

My solution is a vertical blank routine to read the PADDLEs four times, average out the readings, and store the result.

The random pixels problem is a question of checking for a "stylus off" reading. This part I've handled in the basic programming. To save processing time in calculating the X-Y coordinates and checking those out of cursor range, I've added a routine to work them all out and store them in an array during the initialisation.

The programs that follow demonstrate ways of using the Touch Tablet.

Tablet Zero is a demo of using the Touch Tablet as a selection device.

Tablet 8 is a Graphics 8 drawing program. As you will see from this, even with the VBI routine it is painfully slow drawing a continuous line. Part of the problem is that the Graphics 8 screen has a higher resolution than the Touch Tablet, which means you have to Plot and Drawto.

So now the ball is in your court. if you can improve on my ideas, send them in. And if you come up with a program using the Touch Tablet send that in too.

As a final suggestion of a use for the Touch Tablet, how about an alternative one finger keyboard handler for handicapped people?

It covers a smaller area than the keyboard, and all the multiple key inputs could be handled as a cumulative input.

The screen for such a program could look like the one shown in Figure I.

A young handicapped friend of ours has a prototype – and very expensive – speech device that is limited to the number of words that can be stored in its memory and printed on the pad surface. Imagine what could be done with a program such as I've described running with S.A.M. . . .

```
10 REM ******************************
20 REM **       TABLET - ZERO      **
30 REM **TOUCH TABLET DEMONSTRATION**
40 REM **        BY KEN WARD       **
50 REM ******************************
60 GOSUB 210
70 REM  MAIN LOOP 
80 X=HOR(PEEK(203)):Y=VER(PEEK(204))
90 IF X=999 OR Y=999 THEN GOTO 80
100 LOCATE X,Y,COLOUR:COLOR K:PLOT X,Y
110 IF STICK(0)<15 THEN COLOR COLOUR+1
28:PLOT X,Y:GOSUB 150
120 COLOR COLOUR:PLOT X,Y:K=K+128:IF K
>160 THEN K=32
130 GOTO 80
140 REM  PROCESS TRIGGER PRESS 
150 IF X<3 OR X>37 OR Y<10 OR Y>18 THE
N ? "YOU'RE OUTSIDE THE GRID - TRY AGA
IN":GOSUB 190:RETURN
160 IF X=2*INT(X/2) OR Y<>2*INT(Y/2) T
HEN ? "YOU'RE ON A GRID LINE - TRY AGA
IN":GOSUB 190:RETURN
170 H=((X-1)/2)+64:V=(Y/2)-4+48
180 ? "YOU HAVE CHOSEN SQUARE ";CHR$(H
);CHR$(V);"  "
190 FOR DELAY=1 TO 10:NEXT DELAY:RETUR
N
200 REM  TITLE SCREEN 
210 GRAPHICS 0:POKE 752,1
220 ? "↓↓     A DEMONSTRATION OF THE U
SE"
230 ? "     OF THE ATARI TOUCH TABLET"
240 ? "▸↓         BY KEN WARD"
250 REM  INITIALIZATION 
260 POSITION 2,21:? "JUST A MOMENT....
setting up arrays"
270 DIM HOR(229),VER(229),TEMP(9)
280 FOR X=0 TO 228:HOR(X)=INT(X/5.4)-3
290 IF HOR(X)<0 THEN HOR(X)=0
300 IF HOR(X)>39 THEN HOR(X)=39
310 NEXT X:HOR(228)=999
320 FOR X=0 TO 228:VER(X)=24-INT(X/8.8
7)
330 IF VER(X)<0 THEN VER(X)=0
340 IF VER(X)>23 THEN VER(X)=23
350 NEXT X:VER(228)=999
360 REM  VERT. BLANK ROUTINE 
370 POSITION 19,21:? "reading VBI rout
ine"
380 FOR X=1536 TO 1720:READ A:POKE X,A
:NEXT X
390 DATA 104,169,7,162,6,160,10,76,92,
228,173,162,6,201,4,240,26,172,162,6,1
73,112,2,153,163
400 DATA 6,173,113,2,153,167,6,200,140
,162,6,192,4,240,11,76,98,228,160,0,14
0,162,6,76,98
410 DATA 228,216,160,3,185,163,6,201,2
28,240,92,24,109,171,6,141,171,6,169,0
,109,172,6,141,172
```

```
420 DATA 6,136,16,231,160,3,185,167,6,
201,228,240,65,24,109,173,6,141,173,6,
169,0,109,174,6
430 DATA 141,174,6,136,16,231,78,172,6
,110,171,6,78,172,6,110,171,6,173,171,
6,133,203,78,174
440 DATA 6,110,173,6,78,174,6,110,173,
6,173,173,6,133,204,160,3,169,0,153,17
1,6,136,16,248
450 DATA 76,98,228,169,228,133,203,133
,204,76,140,6,0,0,0,0,0,0,0,0,0,0,0,0,
0,104,160,98,162,228,169,7,76,92,228
460 X=USR(1536)
470 POKE 53279,0
480 REM DRAW GRID FOR DEMO
490 POSITION 2,8:? " A B C D E F G H I
 J K L M N O P Q R"
500 ? "┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬
┬┬┐"
510 X=1:FOR A=1 TO 7 STEP 2
520 POSITION 1,9+A:? X;"| | | | | | |
| | | | | | | | | | | |"
530 POSITION 1,9+A+1:? " ├┼┼┼┼┼┼┼
┼┼┼┼┼┼┼┼┼┼┼┤"
540 X=X+1:NEXT A
550 POSITION 1,18:? "5| | | | | | | |
| | | | | | | | | | | |"
560 POSITION 1,19:? " └┴┴┴┴┴┴┴┴┴
┴┴┴┴┴┴┴┴┴┘"
570 K=32:POKE 703,4
580 ? "KOK - NOW USE THE TOUCH TABLET
TO":? "CHOOSE A SQUARE - ":? "PRESS TR
IGGER ON SELECTION"
590 RETURN
10 REM ****************************
20 REM **       TABLET - 8       **
30 REM ** A GR.8 T/TABLET PROGRAM **
40 REM **       BY KEN WARD      **
50 REM ****************************
60 GOSUB 160
70 GRAPHICS 8+16:POKE 709,2:POKE 710,8
:KEN=1
80 X=HOR(PEEK(203)):Y=VER(PEEK(204))
90 IF PEEK(764)<255 THEN GOTO 610
100 IF X=999 AND Y=999 AND STICK(0)<15
 THEN GOTO 740
110 IF X=999 OR Y=999 THEN GOTO 80
120 LOCATE X,Y,COLOUR:COLOR K:PLOT X,Y
130 IF STICK(0)<15 AND (((X)H-4) AND (
X<H+4)) AND ((Y>V-4) AND (Y<V+4))) THE
N COLOR KEN:PLOT H,V:DRAWTO X,Y:H=X:V=
Y
140 COLOR COLOUR:PLOT X,Y:K=K+1:IF K>1
 THEN K=0
150 H=X:V=Y:GOTO 80
160 REM TITLE SCREEN
170 GRAPHICS 0:POKE 709,18:POKE 710,24
180 ? "↓↓↓     T A B L E T  -  8"
190 ? "↓    A GRAPHICS 8 DRAWING PROGRA
M"
200 ? "↓     FOR THE ATARI TOUCH TABLET
"
210 ? "↓↓↓        BY  KEN WARD"
220 REM INITIALIZATION
230 POSITION 9,21:? "setting up arrays
...."
240 DIM HOR(229),VER(229),TEMP(9),FILE
$(15),INPUT$(15)
250 FOR X=0 TO 228:HOR(X)=INT(X*1.5)-1
5
260 IF HOR(X)<0 THEN HOR(X)=0
270 IF HOR(X)>319 THEN HOR(X)=319
280 NEXT X:HOR(228)=999
290 FOR X=0 TO 228:VER(X)=215-X
300 IF VER(X)<0 THEN VER(X)=0
310 IF VER(X)>191 THEN VER(X)=191
320 NEXT X:VER(228)=999
330 REM VERT. BLANK ROUTINE
340 POSITION 9,21:? "reading VBI routi
ne.."
350 FOR X=1536 TO 1720:READ A:POKE X,A
:NEXT X
360 DATA 104,169,7,162,6,160,10,76,92,
228,173,162,6,201,4,240,26,172,162,6,1
73,112,2,153,163
370 DATA 6,173,113,2,153,167,6,200,140
,162,6,192,4,240,11,76,98,228,160,0,14
0,162,6,76,98
380 DATA 228,216,160,3,185,163,6,201,2
28,240,92,24,109,171,6,141,171,6,169,0
,109,172,6,141,172
390 DATA 6,136,16,231,160,3,185,167,6,
201,228,240,65,24,109,173,6,141,173,6,
169,0,109,174,6
400 DATA 141,174,6,136,16,231,78,172,6
,110,171,6,78,172,6,110,171,6,173,171,
6,133,203,78,174
410 DATA 6,110,173,6,78,174,6,110,173,
6,173,173,6,133,204,160,3,169,0,153,17
1,6,136,16,248
420 DATA 76,98,228,169,228,133,203,133
,204,76,140,6,0,0,0,0,0,0,0,0,0,0,0,0,
0,104,160,98,162,228,169,7,76,92,228
430 X=USR(1536)
440 POKE 53279,0:RETURN
450 REM SAVE ROUTINE
460 OFF=USR(1711)
470 OPEN #1,8,0,FILE$
480 TOP=256*PEEK(106):SC=PEEK(88)+256*
PEEK(89)
490 ST=TOP-SC:HI=INT(ST/256):LO=ST-(HI
*256)
500 POKE 850,11:POKE 852,PEEK(88):POKE
 853,PEEK(89):POKE 856,LO:POKE 857,HI
510 T10=USR(ADR("hhh*LV*"),16):CLOSE #
1
520 RESET=USR(1536)
530 GRAPHICS 8+16+32:POKE 709,2:POKE 7
10,8:POKE 53279,0:GOTO 80
540 REM LOAD ROUTINE
550 OFF=USR(1711)
560 OPEN #1,4,0,FILE$
570 POKE 850,7:POKE 852,PEEK(88):POKE
853,PEEK(89):POKE 856,255:POKE 857,255
580 T10=USR(ADR("hhh*LV*"),16):CLOSE #
1
590 RESET=USR(1536)
600 GRAPHICS 8+16+32:POKE 709,2:POKE 7
10,8:POKE 53279,0:GOTO 80
610 REM CHECK KEYPRESS
620 IF PEEK(764)=62 THEN POKE 764,255:
GOSUB 680:GOTO 450
630 IF PEEK(764)=0 THEN POKE 764,255:G
OSUB 680:GOTO 540
640 IF PEEK(764)=31 THEN POKE 764,255:
KEN=1:GOTO 80
650 IF PEEK(764)=50 THEN POKE 764,255:
KEN=0:GOTO 80
660 IF PEEK(764)=33 THEN POKE 764,255:
KEN=0:GOTO 760
670 GOTO 80
680 REM GET FILE NAME
690 GRAPHICS 8+32:POKE 709,18:POKE 710
,24
700 ? "K↓DEV:FILENAME ";:INPUT INPUT$
710 IF INPUT$="C" OR INPUT$="C:" THEN
FILE$="C:":RETURN
720 IF INPUT$(1,2)<>"D:" THEN FILE$="D
:":FILE$(3)=INPUT$:RETURN
730 FILE$=INPUT$:RETURN
740 REM HELP SCREEN
750 IF FLAG<4 THEN FLAG=FLAG+1:GOTO 80
760 GRAPHICS 8+32:POKE 709,18:POKE 710
,24
770 ? " 'S' SAVE SCREEN   | 'L' LOAD SC
REEN"
780 ? " '1' DRAW MODE     | '0' ERASE M
ODE"
790 ? " PRESS TRIGGER TO RETURN TO DRA
WING"
800 IF PEEK(764)<255 THEN 610
810 IF STICK(0)=15 THEN 800
820 FLAG=0
830 GRAPHICS 8+16+32:POKE 709,2:POKE 7
10,8:GOTO 80
```

Atari and Logo – just made for each other

DEREK RADBURN introduces you to a micro love affair

ATARI users who buy Atari Logo are doubly fortunate. Not only have they acquired a programming language which is friendly to use and will help them to develop a good programming style, but also with it they have a version of Logo with some very special features.

The Atari machines are really very good host computers for Logo. Firstly, because the Logo comes on a ROM cartridge and, because of the way the machine is arranged, it really is a Logo machine.

It doesn't waste valuable memory space resetting vectors, as is done with some disc-based Logos running on what are really Basic computers.

Secondly, the Atari's collision counter routine provides an exciting extension to Logo which allows interesting things to be done with the hardware sprites – resident in the computer, as distinct from the program – which are another feature of the machine.

Thirdly, the Atari has four voices and is capable of producing musical effects. Though possession of a disc drive is an undoubted advantage, it is not essential. Your work can be saved on to cassette.

To get the best from this Logo, though, it is highly desirable that you have a colour television – or you may use a monitor with either the 600XL or 800XL. Atari Logo will run on the 400 and 600XL machines which have only 16k of RAM although this really does not leave much room to do a lot.

The whole point about Logo is that it is intended to be a tool to think with. Unlike some other programming languages, it does not try to force the human to accommodate to the machine, but attempts to create a highly-consistent world which is friendly to the human user.

This is reflected in the Logo data types, words and lists, which really reflect the types of objects which human beings process. What are words? Well, in Logo words are pretty much the same as they are in the human world, collections of characters, terminated by a space.

Spaces are significant in Logo, miss them out and the system will issue an error message.

There are a group of words with which the Logo system starts up called primitives. When you type one of these the system knows what to do.

There is a very simple syntax which indicates to the system how it is to treat any words which it encounters. If there is no mark in front of the word – that is just the word itself – it attempts to execute it.

It sees it as a command, either a primitive or a procedure. But more on procedures shortly.

If the word is immediately preceded by a colon, "dots" in Logoese, it means that the value assigned to that word is being called for – that is, it attempts to evaluate.

However, if there is a double quote in front of the word, such as "TOPS, then the system takes the word literally and does not do anything to it. So PRINT"TOPS would result in TOPS appearing on the screen.

Let's expand this a little by attaching a value to TOPS. This can be done by using the MAKE primitive. In order to work, MAKE has to have two things. The first must be a name (quoted word) and the second may be another word or a list. I'll deal with lists in a moment.

Here's an example: MAKE "TOPS "SPOT. This has now assigned the word SPOT as a value to be called when you type PRINT :TOPS, so consequently SPOT appears on the screen.

If you should attempt to reference a value when a word hasn't been assigned one, you will get the message saying FIBS HAS NO VALUE (if FIBS was the name of the word), which is somewhat more helpful than NO SUCH VARIABLE IN 2050.

Numbers are treated by Logo as being special forms of words. Lists are simply collections of words or other lists. They frequently form the values which get assigned to words.

Lists are indicated, delimited, by the use of square brackets, thus [THIS IS A LIST].

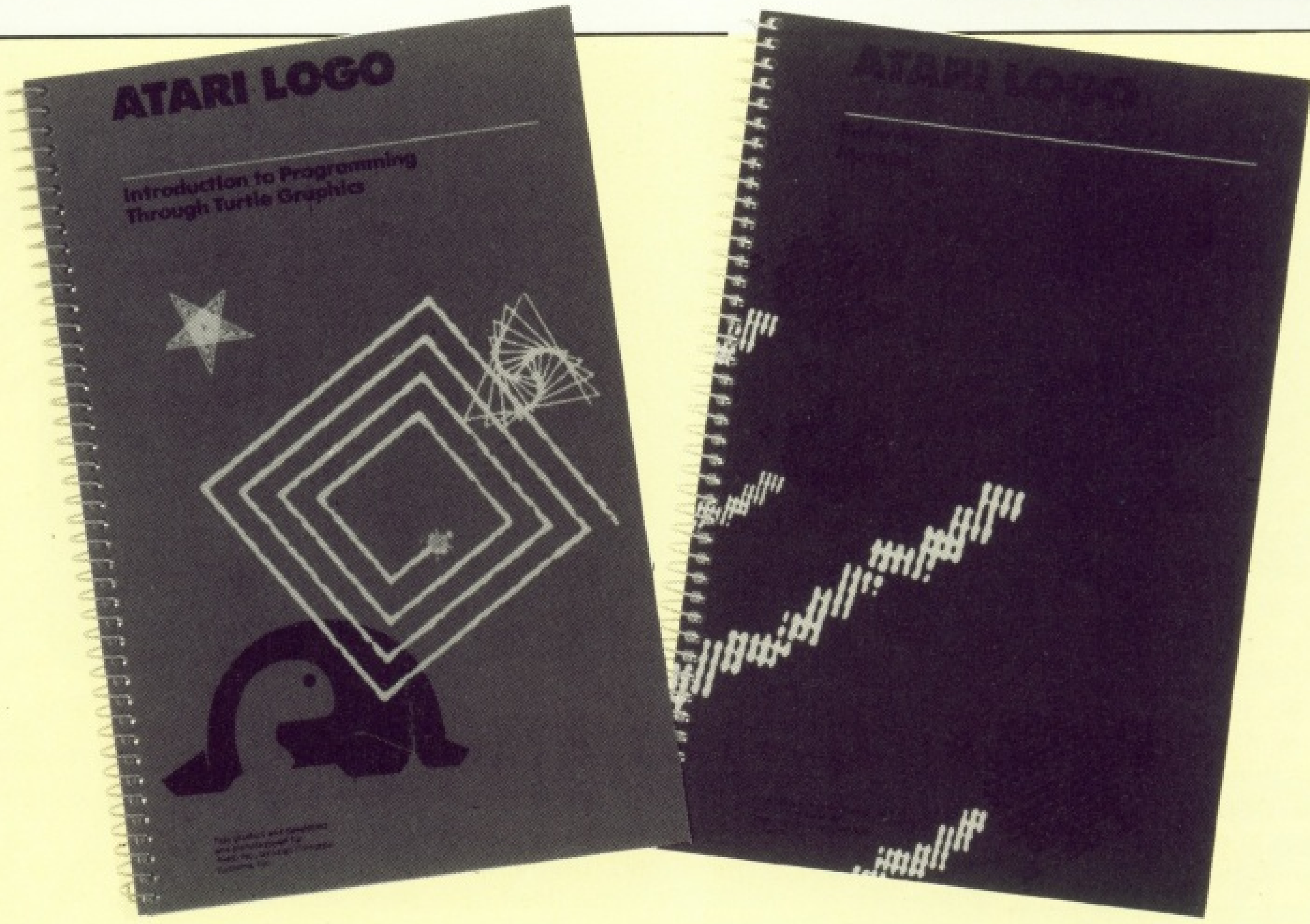
If you MAKE "GREET [HELLO, ATARI TURTLE LOVERS.] and then follow it with PRINT :GREET you should be able to predict now what will appear.

You may have noticed that all of the Logo here has appeared in upper case characters. This is because Logo is case-sensitive and does not recognise lower case letters.

With the earlier Atari 400 and 800 machines it is very important to be aware of this, since the Caps Lower key is directly adjacent to the Return key. Accidentally pressing this key will result in lowercase letters, which will not be recognised by Logo.

By some rather unhappy ergonomics it is necessary to press two keys together – Shift and Caps/Lower – to recover uppercase letters. This feature is particularly unhelpful to young Atari Logo users.

Turtle graphics is one of the best known and copied features of Logo.



Indeed, it is an all too-common misconception that this is all that Logo is.

I hope by choosing to start with words and lists I have dispelled some of these mistaken notions. Essentially, turtle graphics is an example of the friendly human interface in Logo.

We all have bodies, and through them we have a spatial awareness of body geometry. We know which way we are facing, and know our position. We do not use coordinates to guide our movement – we simply go forwards or backwards and make turns which alter our heading at appropriate times.

Logo uses precisely these commands to control the position of the turtle. It may be a robotic device which runs around on the floor or it may be just a symbolic screen turtle. The Atari one actually does look like one, but most are only triangles.

Whichever is used, both can record their paths by putting down a pen, one of three per turtle. There are four turtles. Their shape can be redefined by the user. This is done by calling the EDSH command. This must be followed by the number of the shape to be edited (or created).

Suppose you wish to create shape 1 – you may have up to 15 – then you would type EDSH 1. You would be rewarded by seeing an 8 column by 16 row grid.

Shape definition takes place by moving around the grid using the combination of holding the Control key down and pressing the cursor arrow keys on the right of the keyboard.

The space bar acts as a toggle which, when pressed, fills in empty cells, or clears filled cells, whichever is under the cursor.

There is one point to bear in mind about these user defined shapes. Unlike the original system turtle character they do not alter their orientation to reflect the heading which has been selected. Put simply, this means your planes could be seen flying backwards across the screen, though I prefer multi-coloured flying pigs.

Another special feature of Atari Logo are multiple turtle sprites. It is possible to have up to four turtles, which may have their original shape or be given one defined by the user.

Any of these shapes may be given a velocity by using the SETSP command, which affects the currently active turtle. The speed may be between –200 and 200 (you can guess the effect of a negative input, can't you?).

Do not view the world of the turtle as being separate from words and lists – everything in Logo is based on these. For example, let's draw a shape (?). Type: FD 30 RT 30 FD 30 LT 30 FD 30. When you press Return, provided you remembered the spaces, the five commands should execute.

Now try this: MAKE "WRIGGLE [FD 30 RT 30 FD 30 LT 30 FD 30]. Clear the screen by typing CS and pressing Return. (At this point I shall expect that you already know or have realised the need to press Return.) Now try: RUN :WRIGGLE.

The RUN command in Logo needs to have list of executable items as its input.

Logo has a nice loop structure, REPEAT, which needs two inputs. The first must be a numeric value which tells it how many times to loop. The second must be a list with executable items for it to do.

Try this: REPEAT 3 :WRIGGLE. Do you see the connection between RUN and REPEAT? Now that use of REPEAT was not too interesting was it? Try this: REPEAT 6 [RUN :WRIGGLE RT 180 RUN :WRIGGLE RT 120].

Another way of achieving the same result is to type TO WRIGRAY. As soon as you typed this and pressed Return notice the change.

Look at the prompt. Instead of the usual toplevel (interactive command level) prompt of ?, you will see a >. This signifies that you are in the defining mode and have begun to define a procedure.

The computer no longer responds

→

immediately to what you type, it is storing it and will only execute it when you tell it to. You do this by typing the name which follows TO. You are on the way to defining a procedure.

The change which occurs in the computer's behaviour when the defining mode is first invoked often causes confusion to novices. There are only two ways in which you can leave this mode.

The first is by typing on a line of its own, the word END, in which case the procedure gets defined. The other is to abort the whole enterprise by pressing the Break key.

With the Atari, this does not have the devastating effect it does with some other machines – Reset does that! Should you press the Break key, definition proceeds no further.

So, to continue, type:

```
TO WRIGRAY
 REPEAT 6 [WRIGGLE RT 180 WRIGGLE
   RT 120]
END

TO WRIGGLE
 FD 30 RT 30 FD 30 LT 30 FD 30
END
```

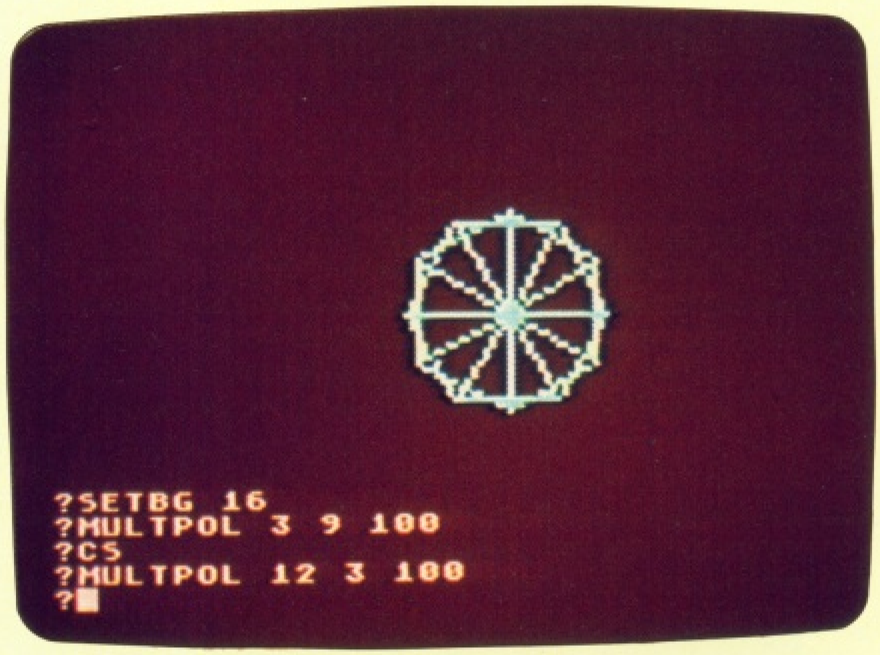
Notice now that we have a variable attached to WRIGGLE and also a procedure called WRIGGLE. The Logo system sees both as different objects.

To alter a previously defined procedure means entering the editor. This can be done by typing ED "WRIGRAY. It is a full screen editor of the sort usually found with Logos.

What you see on the screen is what you get. Movement around the screen is by the Ctrl and arrow key combination already described. After editing, you may leave and retain your amendments by typing Esc, or abandon the changed version, while still retaining the original unaltered version by typing Break.

Logo is essentially an exploratory environment. Although Atari Logo is accompanied by extremely good documentation, the best way to become accomplished with it is to *do* it.

In this article I have purposely avoided giving anything which might be an "end". Rather, I have tried to hint at beginnings for your own learning and pleasure.



Here are some procedures for you to tinker with. Prettypol brings them all together. Experiment with them and have fun.

This procedure draws a regular polygon of a given number of sides, of a size scaled by the number of sides:

```
TO POLYG :SIDES :SIZE
 REPEAT :SIDES [FD (:SIZE / :SIDES)
   RT (360 / :SIDES)]
END
```

This procedure draws a predetermined number of polygons rotated around an axis:

```
TO MULTPOL :TIMES :SIDES :SIZE
 REPEAT :TIMES [POLYG :SIDES :SIZE
   RT 360 / :TIMES]
END
```

This procedure causes three polygonal patterns to be drawn in random colours in random positions. BG gives the value of the background colour, ST shows turtle, FS gives a full screen of graphics – you should be able to deduce the rest. It's important to note that this procedure leaves the system as it found it:

```
TO PRETTYPOL
 MAKE "SCREENCOL BG
 SETBG 6
 ST FS
 REPEAT 3 [PC.CHOOSE CHOOSEPOS
MULTPOL (1 + RANDOM 12) (3 + RANDOM
10) (100 + RANDOM 100) PENCHOOSE]
 HT
 FINISHP SS
 SETBG :SCREENCOL
 ERN "SCREENCOL
END
```

This procedure chooses a new pencolour, but checks to see that it is not the same as the background colour. Notice the IF test. IF is always followed by a list which gets RUN if the condition tested for is found to be true. Optionally a second list will get executed if the condition is false:

```
TO PC.CHOOSE
 SETPC PN (1 + RANDOM 127)
 IF BG = PC PN [PC.CHOOSE]
END
```

This one simply sets a position for the turtle:

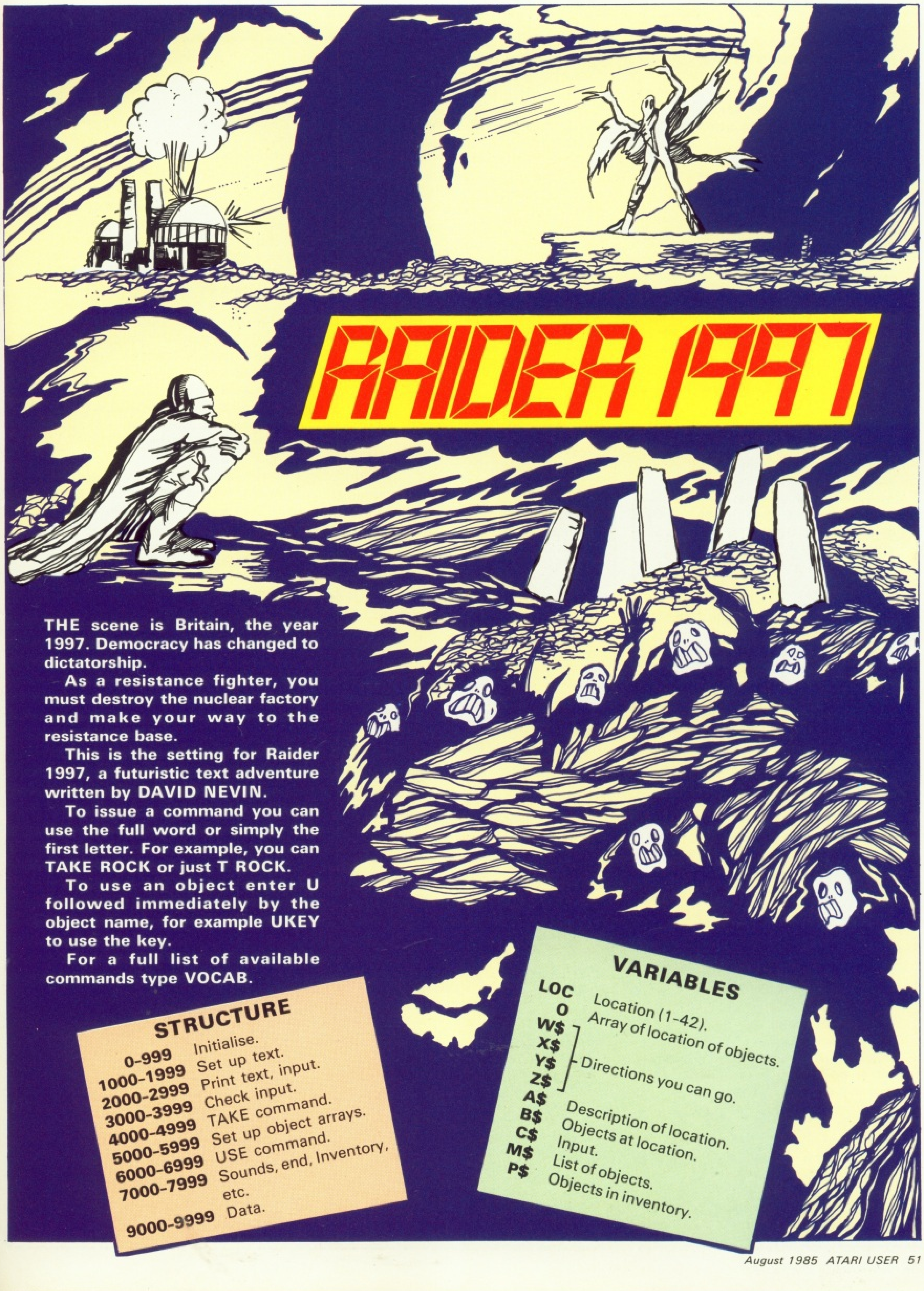
```
TO CHOOSEPOS
 PU
 SETX -50 + (RANDOM 100)
 SETY -50 + (RANDOM 100)
 PD
END
```

This procedure cycles through the three pens:

```
TO PENCHOOSE
 IF PN < 2 [SETPN (PN + 1)] [SETPN
   0]
END
```

This procedure waits for a key to be pressed to indicate the user has finished:

```
TO FINISHP
 IF KEYP [STOP] [RECYCLE FINISHP]
END
```



RAIDER 1997

THE scene is Britain, the year 1997. Democracy has changed to dictatorship.

As a resistance fighter, you must destroy the nuclear factory and make your way to the resistance base.

This is the setting for Raider 1997, a futuristic text adventure written by DAVID NEVIN.

To issue a command you can use the full word or simply the first letter. For example, you can TAKE ROCK or just T ROCK.

To use an object enter U followed immediately by the object name, for example UKEY to use the key.

For a full list of available commands type VOCAB.

STRUCTURE

0-999	Initialise.
1000-1999	Set up text.
2000-2999	Print text, input.
3000-3999	Check input.
4000-4999	TAKE command.
5000-5999	Set up object arrays.
6000-6999	USE command.
7000-7999	Sounds, end, Inventory, etc.
9000-9999	Data.

VARIABLES

LOC	Location (1-42).
O	Array of location of objects.
W$** **X$ **Y$** **Z$**	Directions you can go.
A$	Description of location.
B$	Objects at location.
C$	Input.
M$	List of objects.
P$	Objects in inventory.

Game

```
10 REM ** RAIDER 1997 **
20 REM ** by DAVID NEVIN **
100 LOC=7:FACT=0
110 DIM A$(30),B$(20),C$(10),W$(10),X$
(6),Y$(6),Z$(6),N$(70),G$(10)
120 DIM M$(150),P$(200),O(50)
130 GRAPHICS 0
140 GOSUB 5000
500 ? :? ;"      RAIDER 1997"
1000 Z=0:X=0:B$="":FOR R=0 TO 7
1010 IF O(R)<>LOC THEN 1030
1020 B$=M$(R*10+1,R*10+10)
1030 NEXT R:IF B$="" THEN B$="NOTHING"
1100 ? "    _______________"
1110 D=0:E=0:F=0:G=0:W$="":X$="":Y$=""
:Z$="":RESTORE (LOC*10+9000):READ A$,D
,E,F,G
1120 IF D=1 THEN W$="NORTH "
1130 IF E=1 THEN X$="SOUTH "
1140 IF F=1 THEN Y$="EAST "
1150 IF G=1 THEN Z$="WEST "
1160 IF X=1 THEN RETURN
1170 IF W$="" AND X$="" AND Y$="" AND
Z$="" THEN W$="NOWHERE !"
2000 REM TEXT
2010 ? :? "YOU ARE ";A$
2020 ? "YOU SEE ";B$
2030 ? "YOU CAN GO ";W$;X$;Y$;Z$
2040 IF LOC=4 OR LOC=19 OR LOC=42 OR L
OC=13 THEN Z=1:GOTO 7500
2100 INPUT C$:IF C$="" THEN 2100
2110 IF LEN(C$)<10 THEN C$(LEN(C$)+1,1
0)="          "
2999 REM INPUT
3000 IF C$(1,1)="N" AND D=1 THEN LOC=L
OC+1:GOTO 1000
3010 IF C$(1,1)="S" AND E=1 THEN LOC=L
OC-1:GOTO 1000
3020 IF C$(1,1)="E" AND F=1 THEN LOC=L
OC-6:GOTO 1000
3030 IF C$(1,1)="W" AND G=1 THEN LOC=L
OC+6:GOTO 1000
3050 IF C$(1,1)="T" THEN 4000
3060 IF C$(1,1)="V" THEN GOSUB 7400:GO
TO 2100
3070 IF C$(1,1)="I" THEN GOSUB 7000
3080 IF C$(1,1)="U" THEN 6000
3090 IF Z=1 THEN RETURN
3100 ? ;"YOU CAN'T DO THAT ":GOTO 2100
4000 IF B$="NOTHING" THEN ? "THERES NO
THING HERE!":GOTO 2100
4010 P$(LEN(P$)+1)=B$
4020 FOR N=0 TO LEN(M$)/10-1
4030 IF M$(N*10+1,N*10+10)=B$ THEN GOT
O 4050
4040 NEXT N:GOTO 4100
4050 O(N)=0:B$=""
4100 GOTO 2100
5000 RESTORE 9900
5010 FOR R=0 TO 7
5020 READ N$:IF LEN(N$)<10 THEN N$(LEN
(N$)+1,10)="          "
5030 M$(LEN(M$)+1)=N$:N$(1,10)=N$
5040 READ N:O(R)=N
5050 NEXT R
5060 RETURN
6000 L=LEN(P$):IF L=0 THEN ? "BUT YOU
HAVE NOTHING!":GOTO 1120
6010 FOR N=0 TO L/10-1
6020 IF P$(N*10+1,N*10+9)=C$(2,10) THE
N GOTO 6100
6030 NEXT N:GOTO 6600
6100 IF LOC=2 AND C$="UFOOD     " THEN
 B$="PASSWORD  ":GOTO 6800
6110 IF LOC=4 AND C$="UPAPERS   " THEN
 D=1:GOTO 6800
6120 IF LOC=16 AND C$="UCROWBAR  " THE
N B$="KEY       ":GOTO 6800
6130 IF LOC=27 AND C$="UOLD COINS" THE
N B$="MONEY     ":GOTO 6800
6140 IF LOC=22 AND C$="UMONEY    " THE
N B$="GUN       ":GOTO 6800
6150 IF LOC=30 AND C$="UKEY      " THE
N F=1:X=1:GOTO 6800
6160 IF LOC=39 AND C$="UCAR KEYS " THE
N B$="TOOLS     ":GOTO 6800
6170 IF LOC=37 AND C$="UMATCHES  " THE
N D=1:? "THEY RELEASE YOU":X=1:GOTO 68
00
6180 IF LOC=25 AND C$="UTOOLS    " THE
N F=1:? "THE WIRE IS CUT":X=1:GOTO 680
0
6190 IF LOC=19 AND C$="UGUN      " THE
N F=1:GOSUB 7100:X=1:GOTO 6800
6210 IF LOC=13 AND C$="UDYNAMITE " THE
N GOSUB 7200:X=1:FACT=1:GOTO 6800
6220 IF LOC=42 AND C$="UPASSWORD " AND
 FACT=1 THEN GOTO 7300
6230 IF LOC=33 AND C$="UNET      " THE
N B$="BOOTS     ":GOTO 6800
6240 IF LOC=40 AND C$="UBOOTS    " THE
N D=1:GOTO 6800
6400 IF LOC<>19 AND C$="UGUN      " TH
EN GOTO 7700
6410 IF LOC<>13 AND C$="UDYNAMITE " TH
EN GOTO 7700
6500 ? "NOTHING HAPPENS!":GOTO 2100
6600 ? "YOU DON'T HAVE THAT":GOTO 2000
6800 P$(N*10+1,N*10+9)="         "
6810 IF X=1 THEN GOSUB 1120:GOTO 2020
6820 GOTO 1120
7000 IF LEN(P$)=0 THEN ? "YOU HAVE NOT
HING":GOTO 2100
7010 H=(LEN(P$)/10)-1:IF H<0 THEN H=0
7020 ? "YOU HAVE"
7030 FOR N=0 TO H
7040 ? "   ";P$(N*10+1,N*10+10)
7050 NEXT N:GOTO 2100
7100 FOR N=1 TO 5:SOUND 0,1,0,15
7110 FOR J=1 TO 20:NEXT J
7120 SOUND 0,0,0,0:NEXT N
7130 ? "THE GUARD IS DEAD"
7140 RETURN
7200 FOR I=0 TO 255:SOUND 2,I,0,14:NEX
T I:SOUND 2,0,0,0
7210 ? "THE FACTORY IS DESTROYED"
7220 RETURN
7300 ? "          CONGRATULATIONS
        YOU HAVE COMPLETED THE ADVENT
URE":GOTO 7600
7310 END
7400 ? "     COMMANDS"
7410 RESTORE 9800:FOR N=1 TO 8
7420 READ N$:? N$:NEXT N:RETURN
7500 FOR N=1 TO 3:GOSUB 2100:NEXT N
7510 ? "      YOU HAVE BEEN SHOT AS A
SPY"
7600 ? "          ANOTHER GAME?  (Y/N)
"
7610 INPUT G$:IF G$<>"Y" THEN END
7620 RUN
7700 ? "          YOU HAVE KILLED YOURS
ELF !":GOTO 7600
9000 REM LOCATION DATA
9010 DATA IN THE LIBRARY,0,0,0,1
9020 DATA NEAR A STARVING TRAMP,1,0,0,
1
9030 DATA ON A ROAD LEADING NORTH,1,1,
0,1
9040 DATA AT AN CHECKPOINT,0,0,0,0
9050 DATA BY A BRIDGE OVER A RIVER,1,1
,0,1
9060 DATA IN A DESERTED HOUSE,0,1,0,0
9070 DATA IN AN EMPTY STREET,1,0,1,0
9080 DATA IN A STREET,0,1,1,1
9090 DATA IN A LOOTED SHOP,0,0,1,0
9100 DATA
9110 DATA ON A WINDING PATH,1,0,1,1
9120 DATA BY A RIVER,0,1,0,1
9130 DATA IN THE NUCLEAR FACTORY,0,0,0
,1
9140 DATA IN A HOUSE,0,0,1,0
9150 DATA
9160 DATA BY A BOX,1,0,0,0
9170 DATA AT A CROSSROADS,1,1,1,1
9180 DATA IN A SCRAPYARD,0,1,1,0
9190 DATA BESIDE A GUARD,0,0,1,1
9200 DATA
9210 DATA IN THE VILLAGE,1,0,0,1
9220 DATA IN A GUNSHOP,0,0,0,1
9230 DATA ON A MOTORWAY,0,0,1,1
9240 DATA IN A HUT,0,0,0,1
9250 DATA BY A WIRE FENCE,0,0,0,1
9260 DATA
9270 DATA IN AN ANTIQUE SHOP,1,0,0,0
9280 DATA IN A VILLAGE,1,1,1,0
9290 DATA ON A MOTORWAY,0,1,1,1
9300 DATA OUTSIDE A LOCKED DOOR,0,0,0,
1
9310 DATA IN THE FOREST,1,0,1,1
9320 DATA IN THE FOREST,1,1,0,1
9330 DATA BY A LAKE,1,1,0,1
9340 DATA IN THE MOUNTAINS,1,1,1,1
9350 DATA ON A HILL,1,1,1,0
9360 DATA IN A QUARRY,0,1,1,0
9370 DATA CAPTURED BY THE FIREWORSHIPE
RS,0,0,0,0
9380 DATA IN THE FOREST,1,1,1,0
9390 DATA BY AN ABANDONED CAR,1,1,1,0
9400 DATA BY A MARSH,0,1,1,0
9410 DATA BY A RAILWAY TUNNEL,1,1,0,0
9420 DATA IN THE RESISTANCE BASE,0,0,0
,0
9800 DATA NORTH ,SOUTH ,EAST ,WEST ,TA
KE,U(OBJECT),VOCAB,INVENTORY
9900 DATA PAPERS,1,MATCHES,14,NET,6,FO
OD,9,OLD COINS,12,CROWBAR,18;DYNAMITE,
24,CAR KEYS,29
```

Silicon Chip

No. 1 FOR ATARI*

Atari 520 ST

GUARANTEE
Our EXCLUSIVE Direct Exchange Guarantee is available FREE OF CHARGE to all our customers.
If a product purchased from SILICON CHIP becomes faulty due to a manufacturing or component defect, within ONE YEAR of purchase, we will, subject to availability, exchange for new.



PRICES
ST PACK INCLUDES:
520 ST
½ Meg. Floppy Drive
Hi-Res B/W Monitor
Plus software listed below
£749.99
Basic, Logo, GEM write, GEM paint and BOS Operating System

Final prices and specs may be subject to change.

POWER WITHOUT THE PRICE
Chipsoft Integrated Accounts Package:-
Including Stock Control, Nominal, Sales and Purchase Ledgers **£250**

ST to Centronics Printer Lead
£29.99
ST Midi Lead
£24.99

SPECS
512K RAM.
Expandable Rom.
MIDI Interface. RS232 Serial/Centronics Parallel Interfaces. Floppy Disk and Hard Disks Interfaces. RGB Video and TV outputs. Sound Generator with 3 channels.
2 Joystick Ports. 3 Graphics and Text Modes –
320×200 = 512 colours, 640×200 = 4 colours, 640×400 B/W.
16 Bit Motorola 68000 Microprocessor running at 8 mHz.

SILICON CHIP LTD, Showroom address:
302 High Street, Slough, Berkshire. Tel: 0753 70639

* *1st Atari Business Centre.*
* *1st Dealer for 130 XE.*
* *1st Dealer for 1029 Printers.*
* *1st Dealer with our own Software House for Software Support.*
* *1st Dealer for 520 ST*

Please phone or write for full details.
(Dealer Enquiries Welcome)

MIKE BIBBY gives you the lowdown on . . .

The inside story of binary operations

IN previous articles we've seen that binary numbers can be added and subtracted just as our more familiar decimal numbers are. And, of course, we can multiply and divide them.

There are, however, other ways of combining two binary numbers that are extremely useful in dealing with computers. They're also easy to use, so let's have a look at them.

Firstly, we'll see how we can NOT a binary number – simple, one-bit numbers first. By the way, we're going to be dealing exclusively with binary numbers this month, so we can drop the % sign.

The rules for doing a NOT are simple:

If the bit is 1 then it becomes 0
If the bit is 0 then it becomes 1

If you like, the NOT converts a bit into its opposite.

So **NOT 1 = 0**
And **NOT 0 = 1**

Why do we use the word NOT? Well, mathematicians often use the number 1 to mean true and 0 to mean false.

So NOT 1 means not true, which means false, which is 0. That is, NOT 1 is 0. And, as not false is most certainly true, NOT 0 is 1.

If we are to NOT a binary number consisting of several bits, we simply apply the rule for NOT to each bit individually.

So **NOT 10110010**
becomes **01001101**

Some people think of this process as turning the number on its head, so it's sometimes called inverting. Others call it taking the complement of the number.

NOT just works on a single binary number. However, there are other sums or operations that have a set of rules for combining two binary numbers.

For instance, we can AND two binary numbers. Let's look at the rules for ANDing a single bit with another bit.

When you think about it, there are four possible combinations of bits that we could AND – 0 with 0, 0 with 1, 1 with 0 and 1 with 1.

We write that we are ANDing, say, 0 with 1 as 0 AND 1.

The rules for ANDing are:

0 AND 0 = 0 (case a)
0 AND 1 = 0 (case b)
1 AND 0 = 0 (case c)
1 AND 1 = 1 (case d)

Notice that the only time the result is 1 – true – is when the two bits ANDed are both 1 – true. This helps us to see why we use the word AND to describe the operation.

If you think of the first bit as "this" and the second bit as "that", what we're doing when we're ANDing is asking whether "this and that" is true.

"This and that" can only be true when both "this" is true AND "that" is true – hence the use of AND to describe the process.

For example, consider the statement that it is dry and sunny.

This is true only if dry is true and sunny is true – case d.

If either of the two, or both are false – cases a, b, c – the whole statement is false, since it isn't both dry and sunny.

We can AND pairs of binary numbers of more than one bit – just apply the rules of ANDing to each bit individually.

For example:

AND 10010110
AND 10110011
gives **10010010**

We can also OR two binary numbers. The rules for ORing a single bit with another bit are as follows. Again there are four possible combinations:

0 OR 0 = 0 (case e)
0 OR 1 = 1 (case f)
1 OR 0 = 1 (case g)
1 OR 1 = 1 (case h)

In this case you only get a false result, 0, when both bits are false. If either or both bits are true, 1, the result is true. It's easy to see why we use OR to describe this. If one OR the other OR both is true the whole thing is true.

Let's use the meteorological analogy again. Consider the statement that it is dry or sunny.

This is only false when it is NOT dry and NOT sunny – case e – otherwise it is TRUE – cases f, g, h.

To sum up, with OR the whole thing is true if either or both the things being ORed is true.

As we did with AND, we can OR pairs of numbers with more than one bit – we just apply the rules of ORing to each bit individually.

For example:

10010110
OR 10110011
gives **10110111**

● *In the next article we'll look at EOR and the use of masks.*

DELVING INTO THE ATARI'S INNARDS

I BOUGHT an Atari 800XL soon after they became available – in the firm belief that the full hardware details would be available, as they were for the earlier Atari 800.

The questions I particularly wanted the answers to are the following:

- *How to enable the shadow RAM. Your article on the XE indicates that the technique used is not as easy for the XL.*
- *The details of the signals and the timing at the edge connectors.*

With this information available, constructors could attach a variety of peripherals to the XL series. – **Donald W. Sharp, Newcastle upon Tyne.**

● The shadow RAM under the Operating System and Basic – that's $A000-$BFFF for Basic and $C000-$CFFF & $D800-$FFFF for the OS – can be switched in and out by changing bits 0 and 1 in location $D301, as shown in the table with the article in the June issue of *Atari User.*

However, it can't be done from Basic, as the computer would instantly crash – hence the warning not to try it with a POKE.

Switching either Basic or the OS *must* be done with a machine code program in control. To stop the OS running before banking it out, set NMIEN ($D40E) to zero. Only set it back to $40 – 64 decimal – when you have switched the OS ROM back into place.

$D000-$D7FF can never be used as RAM, because they contain information vital to other chips such as Antic, Pokey and Gtia.

There was a very good series of articles about using the Parallel bus in the January to April Antic Magazine (Vol. 3/9 to Vol. 3/12).

It started from scratch, and ended up showing you how to build and run an RS-232 interface via the bus connector.

Bulletin boards

THANKS for an excellent and much needed magazine for the Atari community. Your June issue on communications was especially interesting.

I saw the numbers for bulletin boards and wondered whether you would include mine in future listings?

The board is Atari based and orientated, called CyberZone, and the number is 01-638 2034. It's a 24-hour auto-answer service. – **Brian Saunders, via Prestel.**

Micro connection

I WOULD like to know whether two Ataris can be connected by an I/O cable (Program Recorder/Disc Drive cord) and a program loaded from one to the other?

Secondly, how do you connect several peripherals which all use the I/O Interface to your Atari ? – **Craig Brady, age 14, Bristol.**

P.S. I typed in the Poke to suppress the clicking sound on my Atari 400 16k, and after trying four times, it still didn't work.

● A standard I/O cable will *not* allow data to go from one machine to another. By re-wiring "Data-In" to "Data-Out", changing the +5V line and re-wiring the clock in/out lines, it *might* work. Can't say we've ever tried it, though.

The easiest way is to save on to tape from one machine, swap the recorder over, and load it back again on the other. Also, the joystick ports can be re-programmed for direct data exchange.

Perhaps someone out there has done it and would care to write in?

There are two I/O sockets on each peripheral in the Atari range, and you simply plug the second unit into the back of the first, and so on.

The codes listed in the letters page in Issue 1 from David Eckersley work OK, but they are for the XL and XE ranges only. They use some of the new features not available on the old Atari range. Sorry.

Edge connectors

I OWN an Atari 400 and would like to explore its expansion and interfacing capabilities. Due to the distinct lack of an expansion orifice, I am finding it not at all easy.

I have a project in mind which would need me to access both the data and address bus. Is there any way I can get to these?

I have noticed some edge connectors on the 400's board – could you explain these?

Also, could you tell me if the Atari 400 is directly compatible with the Atari disc drive, without the expansion interface? – **Edmund McConnell, Leicester.**

● Information on circuit layout, connections, and so on is in the "Technical Reference Notes for the 400/800", published by Atari at £17. It

→

Jumping into difficulties

I HAVE endeavoured to type the Frog Jump game – in your June issue – into my computer.

However, this keeps coming up with Error on Line 340, 350 and 360.

I haven't typed any further, so do not know if there are any other errors.

I am wondering, therefore, if you could advise me of the correct lines for this game.

Also, I have an Atari 800 with cassette, but am thinking of buying a disc drive. Can you tell me if it is possible to record from the cassettes to disc?

I have many games on the cassettes, but don't want to buy a disc drive if it is impossible to break into the program – **G. Newin, Walton-in-Thames.**

● There were no errors in the listing as printed, so you have almost certainly made some typing mistakes.

You don't say what the error number was, but we suspect it was an Error 5. If this is the case, you should check Line 60 very carefully and make sure you have entered it as listed.

For example, make sure you have used 1 – number one – and not I – capital letter i – in strings like L1$.

Most commercial games are fairly heavily protected, so you would probably not be able to move them to disc – particularly if they load in more than one section.

However, the time saved in loading possibly outweighs the cost of replacing your favourite games.

You may also be able to recover some of the money by selling the cassette versions to people who haven't yet got a disc drive.

ATARI USER Mailbag

WE welcome letters from readers – about your experiences using the Atari micros, about tips you would like to pass on to other users . . . and about what you would like to see in future issues.

The address to write to is:

Mailbag Editor
Atari User
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY

should be available from specialist shops, or mail order from Software Express, Silica Shop, and others.

The edge connectors you mention are extensions of the processor bus – left in the machine as engineer test points, but it's up to you what you want to do with them.

Don't forget that if you open your machine, you will invalidate your guarantee.

All Atari peripherals – disc drives, printers, cassette recorders, and so on – will connect to any Atari computer directly, via the 13-pin socket on the side.

However if you only have a 16k machine, you would only have about 8k left for programming after DOS loads into memory.

Why the bleeping?

WHEN I got my 800XL I found there was a "bleep" on the screen every third line you went down.

I went to the shop where I got it from and they checked it. It seemed that the "bleep" was on all of the 800XLs.

So please, please tell me what is that "bleep" doing there?

Is it some kind of safety device? – **Neil McCulloch, Denny, Stirlingshire.**

● We're not sure what you mean but suspect that you're referring to the beep which warns that you are about to exceed the allowable length of program line. If so, it's certainly meant to happen.

Memory check

I HAVE a 600XL. Being only 16k memory, many advertisements for games and utilities don't help me at all.

If I want to order anything I first have to write or telephone the company selling the product to see how much memory it takes up.

I am saving up for the 48k expansion, but for all the people who have only got 16k and will stay with 16k could you please ask your advertisers to show how much memory the product needs?

I know that the 600XL is "going out" but don't rub salt in the wounds.

Also regarding the monthly disc – spare a thought for those with no disc drive and put the offer on cassette as well. – **K. Fallas, Middleton, Manchester.**

Doing it the hard way

CONGRATULATIONS on your magazine.

I have a 600XL which I bought in January and until Atari User was issued I had learned practically nothing.

The articles by Dave Russell, Mike Bibby, Pete Bibby and others are so easy to follow that I'm amazed by what I can do already.

I have the books Atari Programming with 55 Programs by Linda M. Schreiber and Making the Most of Your Atari by Paul Bunn, and although I learned a little bit from these I wasn't learning enough.

I could never figure out what data was until I read Dave Russell's second article on graphics mode. Now I can write small programs using data.

I hadn't got a clue what binary was until Mike Bibby's article on the subject. (I've forgotten for the moment how to write J). Now I can convert numbers into binary.

Now about the listings. I tried four times to write (if that's the term) Alphabet Train into my computer, which took me about three hours each time.

I could never get it to work, so I tore it up in disgust.

I did manage to get Frog Jump to work, but not very well – but that could be because of my joystick.

Submarine works fine, but not Etcha Sketch nor Attack Squash.

I did have a program recorder at one time but could never get it to record any of the listings so I got rid of it.

Now all I have is my computer and joystick, so I have to type in the listings every time – so it would be great if I could get them to work every time.

Any tips please? – **Colm Keegan, Holyhead.**

● Our main suggestion is that you buy another recorder or save up for a disc drive. It will save you hours of typing time which you can use to debug your programs.

Defender score

COULD you tell me if there is an Atari user group in Bristol?

Also, having seen that high score on Drop Zone, I would like to see whether anyone can beat my top score on Defender. I managed to reach over 3 million and gave up with over 60 lives – although it took me about 2½ hours.

Let's have more articles and programs on the more-complicated side of the Atari computer – machine language techniques, hardware, and so on.

Also, is it possible to get a modem which costs less than £100? – **P. Fragapane, Bedminster, Bristol.**

● The address of the East Bristol user group is c/o 2 Channons Hill, Industrial Estate, Fishponds, Bristol.

The Maplin modem costs around £50 . . . but you'll need an 850 module as well. This applies to all the cheaper modems as far as we know.

Frightened off

YOUR news item about Atari's going "bump in the night" seemed quite appropriate considering my 800XL's penchant for "locking-up" when I play the Scott Adams' Ghost Town adventure.

I wonder if other readers have found that their machines take fright in this manner? – **J. Hugill, Leicester.**

Checking errors

I WISH to comment on Les Bostock's request that some form of typing check program be included in your magazine.

I run a computer club for 5 to 13-year-olds and when I see good programs in magazines I ask the children if they would like to type them in. They do, and enjoy it.

When all those lines have been typed in and they then try to run it, you can imagine how disappointed they are when it shows errors at line so and so.

It is then left to me to try to debug the programs. This takes up much-needed time when I could be doing something else for the club.

So yes I do so agree with

Les that an error-checking program would be of great assistance.

It's a shame that all Atari support magazines could not use the same error-checking programs, something like Typo II, which I think is the best, in trying to assist their readers.

May I say that you have a good magazine. – **B. Spooner, Fishguard, Pembs.**

Confusing check-sums

I HAVE just bought Issue 2 of your great magazine and am pleased to say that it is even better than Issue 1.

I find Bit Wise very useful, but how about an article or three on assembly/machine code?

As for the argument over a check-sum routine, my views depend on which type you decide to offer, as the typo tables for Antic and so on are quite confusing. A better type is the one used by Compute.

You argued that it's good to get some practice at de-bugging, but if you are inexperienced you might find this very difficult and having a check-sum would reduce the frustration of typing in listings.

Also, if you do want to de-bug you have not got to use the check-sum if you don't want to.

Finally, as many big American magazines use typo tables it must be of some use to American users, and if the typo tables are not really used you can always drop them at a later date.

Another article that would be welcome is an explanation and programs showing the use of player/missiles and redefining the character set as this is ignored in my 800 manual and I have not discovered any books explaining them simply. – **N. Buckle, Crayford, Kent.**

● It's interesting that you don't like Antic's check-sum methods – many people have suggested we use the same method as Antic.

Atari's on the air

CONGRATULATIONS on a brilliant magazine.

I am a radio amateur and have been using – or trying to use – my Atari 800 for radio-based programs.

I have been able to send radio teletype and Morse in both transmit/receive and also I have a very good Morse training program.

Unfortunately, whenever I approach software suppliers about programs for radio communications, all I get is: "Atari? That's a games machine – we don't keep anything like that".

As this type of software is readily available for such machines as BBC, Spectrum, Dragon, Commodore, Amstrad, to name but a few, what has Atari got that these other machines do not seem to have? Could it be lack of support?

Anyway, if you know anybody who can help, or anybody who wants help, or anybody who is just interested with radio-type programs then can you please pass on my name? – **J.M.A. Sheppard, Bristol.**

Typing error

I'M a beginner with the Atari and so your magazine has taught me a great deal.

But unfortunately, with quite a few games which I have typed in, when I Run it always produces an error.

I always check the listing so it isn't a typing error.

I typed in Attack Squash and it produces error at line 830. It said goto 720 which doesn't even exist.

Could you please help me to understand my computer, and explain to me the error in Attack Squash.

Also, could you please tell me whether there are any groups around the Orpington area, where I could go to talk to other people with the same problems? – **G. Gouveia, Orpington, Kent.**

● We reprinted Attack Squash as it appears in the Atari Book of Games. You're right that line 720 doesn't exist, but the author obviously altered the program so that it never gets to line 830. Hence you must have made a typing error somewhere along the line.

There is a user group in Tunbridge Wells and the contact is Mr T. Chamberlain, 29 Albany Hill, Tunbridge Wells, Kent TN2 3RX.

Switch-on sequence

I HAVE an Atari 600XL and my dad has a Tandy TRS-80 colour computer.

In one of the Tandy manuals it says that switching on the computer without connecting it to the television can damage it. Why is this the case?

Also, does it apply to Atari micros, as I have a friend with an Atari 800XL which he leaves on for long periods of time with the television switched off.

It says in one of the Atari manuals that inserting or removing a cartridge with the computer switched on can damage the cartridge. Can it also damage the computer? – **Peter Goulden, Helston, Cornwall.**

● As far as we know switching a micro on without connecting it to a TV can't do any damage.

Possibly what your Dad's Tandy manuals are suggesting is that you turn on all the peripherals – TV, disc drives etc – before you turn on the micro. This will stop a possible mains "spike" from something like the TV on-off switch damaging your computer.

For the same reason, you should turn the micro off first. Plugging or removing cartridges with the computer switched on could cause damage and should be avoided. It's a bit like surgery without anaesthetic – possible, but not recommended!

Games shortage

YOUR magazine is a most helpful teaching aid into computer programming.

As for the Atari company, I am not full of its praise.

On purchasing the Atari 800XL I was surprised that Atari did not supply at least one games tape to test the machine.

And the manuals that came with the machine contained typing errors.

The main problem concerned the computer, which would not load. But as an amateur, I had no idea what was the matter.

I first decided that the tapes were at fault. I had these tested – they were all right.

So I changed the Atari 1010 tape recorder for another one. Still no luck.

In the end I received a new computer but this would not load either.

This time the Atari tape recorder was at fault. It is now being repaired.

Throughout this whole episode there was no Atari dealer to ask advice from.

There is also the added difficulty of finding Atari computer games.

Woolworths of Hanley and Wolverhampton have depleted their stock and inform me that they are no longer interested in stocking Atari games tapes.

This also applies to all W.H. Smith shops. They only sell Atari books and Laskys do no better.

On reflection, I would have done far better buying Commodore, or Spectrum,

At least they sell their own products through their shop, with well trained staff, and not through individual electrical outlets. – **H. Smith, Stafford.**

● If software is difficult to find, why not contact some of our advertisers?

FREE T-shirt for all new



Here's a really unbeatable offer for all Atari users!

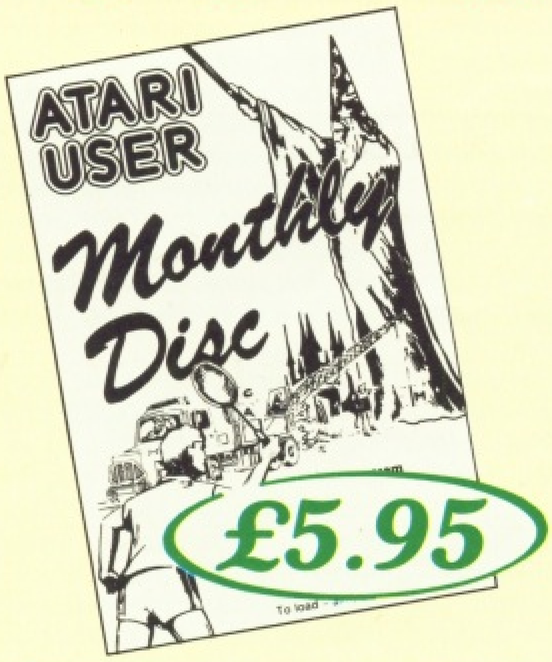
This top-quality T-shirt, woven in an attractive shade of grey with the Atari logo in red, is a genuine American 'Fruit of the Loom' product, made from 50% cotton/50% polyester.

Worth £4.25, it will be sent FREE with every new subscription ordered on the form on the right!

The T-shirts are also on sale at £4.25. Please order on our official order form.

Made in three sizes:	
Small	– 34"-36"
Medium	– 36"-38"
Large	– 38"-40"

Keying in long programs too much of a chore?



Then give your fingers a rest by sending for our monthly disc, containing all the programs from each issue of Atari User. See order form opposite.

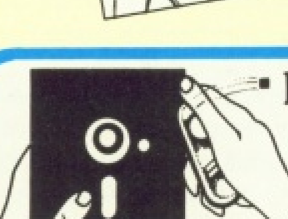
AUGUST: Assembler: Make machine code programming easier. **Fruiti Gambler:** Save money with this fruit machine simulation. **Mandala:** Complex patterns made easy. **Protection:** Routines to protect your programs from prying eyes. **Display List:** Demonstration programs. **Raider 1997:** Futuristic text adventure. **Touch Tablet:** Demonstration programs.

JULY: Bomb Run: Flatten the deserted city and land safely. **Disassembler:** Find out what's going on deep inside your Atari. **Treasure Hunt:** Use logical thinking to find the treasure. **Password Generator:** Keep generating passwords till you find one you like. **Keyboard:** Convert your micro into an organ. **Quasimodo:** Can you sort out the mess of ropes in the belfry?

JUNE: Frog Jump: Guide the frog across the road and river to his home in this version of the arcade classic. **1300XE Ram Power:** Use the extra 64k of memory to good effect, or use the drawing routines to produce some pretty displays. **Submarine:** Scuttle the submarines and practise a little coordinate geometry at the same time. **Etcha-Sketch:** Draw pretty pictures with only a joystick. **Random Numbers:** Get random numbers from machine code. **Filthy Fifteen:** Can you keep the Filthy Fifteen happy in their cells?

MAY: Alphabet Train: The combination of colour, sound and animation makes this early learning game a winner with the children. **Sounds Interesting:** Drive your neighbours potty with these ready-made sounds. **Hexer:** Enter, display and run machine code programs with this hexadecimal loader. **Attack Squash:** A fast-action game to keep you on your toes. **Reaction Timer:** See how fast your reactions really are. **Binary:** Use this program to convert denary numbers to binary notation.

Double the capacity of your discs with this money-saving offer!



DISK DOUBLER

Allows you to cut out a correctly positioned notch which will enable you to use BOTH sides of the disc – and HALVE your costs. Extremely well made to a compact design, it has a unique position guide to assure pin-point accuracy!

£9.95

Free disc with DOS 2.5 with each order

w subscribers to ATARI USER

Back issues



£1.25

May issue: Profile of Jack Tramiel, preview of the new machines, Attack Squash, Adventuring, Alphabet Train, Hexer utility, Software reviews, Sounds, the 6502, Microscope, Atari Insights – regular series of tutorials: Bit Wise, Beginners and Graphics, PLUS News and Mailbag.

June issue: In-depth analysis of the 130XE, Submarine, Adventuring, Random numbers, Software reviews, Frog Jump, Microscope, Sounds, Atari Insights – regular series of tutorials: Bit Wise, Beginners and Graphics, special 12 page feature on Communications, PLUS New and Mailbag.

July issue: Disassembler, Bomb Run, DOS 2.5, 17 Commandments, Adventuring, Display List Tutorial, Software Reviews, Power Functions, Treasure Hunt, Keyboard Sounds, Microscope, Insights – Regular series of tutorials: Bit Wise, Beginners and Graphics PLUS News and Mailbag.

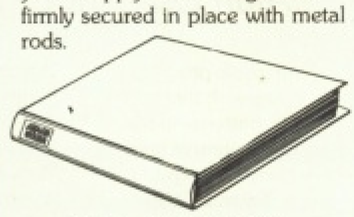
Proctect your 130XE with our luxury dust cover made of soft, pliable, clear and water-resistant vinyl, bound with strong cotton and decorated with the magazine's logo.



Only £3.95

How to keep your collection complete

Bound in chocolate brown pvc and bearing the Atari User logo, this handsome binder will hold a year's supply of the magazines firmly secured in place with metal rods.



Only £3.95 (UK)

Look what's coming in Atari User!

- ★ Lots of easy-to-follow features on everything to do with Atari computing. Whether you're a beginner or an expert, you'll always find something to delight and intrigue you.
- ★ In-depth independent evaluations of all the new hardware add-ons now being developed to make your Atari much more powerful and much more versatile.
- ★ Reviews of all the very latest games, educational and business programs now being produced for the Atari.
- ★ PLUS lots of listings you will be able to key in yourself – games, utilities, graphics.

ORDER FORM

All prices include postage, packing and VAT and are valid to September 30

Please enter your requirements by ticking boxes (✓) £ p

Annual subscription
- UK £12 — 7001
- EIRE £13 (IR £16) — 7002
- Overseas (Surface) £20 — 7003
- Overseas (Airmail) £40 — 7004
- *Free T-shirt – circle size required* — S/M/L

Commence with ______ issue TOTAL ______

Back issues £1.25 UK, £1.50 Overseas
- May 1985 — 7005
- June 1985 — 7006
- July 1985 — 7007

TOTAL ______

Monthly Disc £5.95
- May issue — 7017
- June issue — 7018
- July issue — 7019
- August issue — 7020

TOTAL ______

DOS 2.5 £1.50 — 7052 TOTAL ______

Miracle Modem £195 — Including software and Atari interface — 7033 TOTAL ______

T-Shirt £4.25
- Small — 7047
- Medium — 7048
- Large — 7099

TOTAL ______

Dust Cover (130XE only) £3.95 — 7031 TOTAL ______

Binder £3.95 UK, £5.00 Overseas — 7029 TOTAL ______

Disk Doubler + *free DOS 2.5* £9.95 — 7051 TOTAL ______

Overseas postage (surface) add 50p per item unless otherwise stated.

Payment: please indicate method (✓) TOTAL ______

☐ Access/Mastercard/Eurocard

☐ Barclaycard/Visa

☐ Cheque/PO made payable to Database Publications Ltd.

Name ______

Address ______

Signed ______

Send to: *Atari User*, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

(No stamp needed if posted in UK) *Please allow 28 days for delivery*

YOU CAN ALSO ORDER BY PHONE: 061-480 0171

Don't forget to quote credit card number and full address.

AT8

ADVERTISERS INDEX



ATARI ST

520ST

POWER WITHOUT THE PRICE

THE NEW ATARI 520ST

Under the new leadership of Jack Tramiel (former boss and founder of Commodore Business Machines), Atari Corporation have marked their entry into the world of business/personal computers with a machine which leaves the competition standing. Tramiel's slogan 'Power Without the Price' has been implemented in the manufacture of the new 512K Atari 520ST colour computer which offers the user amazingly high performance at an incredibly low price. Launched as a work-station, this new system incorporates seven software packages as well as the 520ST computer with 512K RAM, mouse controller, high resolution monochrome monitor (640x400), 95 key keyboard (with 18 key numeric keypad), MIDI interface, GEM and a 500K 3½ inch disk drive, all for the package price of only £651.30 (+VAT = £749). Dubbed the 'Mac beater' and the 'Jackintosh' (after Atari's Chief, Jack Tramiel), Atari's new machine has been directly compared with the Apple Macintosh RRP £2595 (+VAT = £2985) which offers similar features and capabilities but at a much higher price. Favourably reviewed by the UK's highly critical specialist computer press, the 520ST is likely to make a great impact in this country as a sophisticated alternative to an IBM PC, APRICOT or APPLE MACINTOSH. Unlike its overpriced competitors, the Atari 520ST can be linked up to a colour monitor to unleash a choice of up to 512 colours. The addition of colour brings out the full potential of graphics packages such as GEM.

USER FRIENDLY GEM OPERATING SYSTEM

The power of the ST is harnessed and made user friendly by the new operating system 'GEM' from Digital Research. GEM stands for Graphics Environment Manager and allows a user friendly colour or B/W graphics interface which closely resembles that of the Macintosh. This similarity extends to the use of moveable resizeable windows, icons to represent objects such as disks and disk drives, and the use of pull down menus and a mouse. The advantage of all this is that the computer becomes extremely easy to use. GEM has now been implemented for the Acorn, ACT, Atari, IBM, ICL, and Olivetti. Software written for GEM on one computer should also run under GEM on another computer. This will enable the market to quickly produce a large library of standard interchangeable software.

FREE SOFTWARE AND FUTURE EXPANSION

The Atari 520ST comes supplied with seven free software packages as listed below: 1) TOS -Tramiel Operating System based on CPM 68K. 2) GEM Graphics Environment Manager by Digital Research (DR) giving a WIMP (Window, Icon, Mouse, Pull down menu) environment. 3) DR GEM Paint for creating graphics masterpieces. 4) DR GEM Write for word processing. 5) Logo learning language to enable you to write your own programs easily using turtle graphics. 6) DR Personal Basic a powerful user friendly version of the Basic programming language. 7) BOS operating system giving you access to dozens of business applications packages already available on the market. Designed with future expansion in mind, the ST also features a host of different interfaces to the outside world and an impressive list of accessories is planned. Atari will soon be releasing a 1000K (1MB) 3½ inch disk drive, and a 15MB hard disk storage system as well as a mass storage compact disk (CD) player capable of storing an entire 20 volume encyclopedia on one disk. A full range of inexpensive printers are planned including dot matrix, daisywheel and thermal colour printers. With its unbeatable graphics, speed and software at a price which is far below that of any comparable personal computer currently on the market, the ST is all set to do battle with the competition. To receive further details of the ST from Silica Shop, just fill in the coupon below with your name and address details and post it to us.

Silica Shop Price: £651.30 + £97.70 VAT = £749.00 This price includes:

- ***512K RAM***
- ***MOUSE***
- ***GEM***
- ***B/W MONITOR***
- ***500K 3.5" DISK DRIVE***
- ***KEYBOARD (95 KEYS)***

£749

ATARI 520ST SPECIFICATION

MEMORY
512K RAM (524,288 bytes)
16K ROM expandable to 320K
Port for addnl 128K plug-in ROM cartridges
200K TOS operating system

GRAPHICS
Individually addressable 32K bit-mapped screen with 3 screen graphics modes:
320x200 pixels in 16 colours (low resolution)
640x200 pixels in 4 colours (med resolution)
640x400 pixels in monochrome (high res)
16 shades of grey in low res mode
512 colours available in low/medium res
8 levels of each in red, green and blue

ARCHITECTURE
4 custom designed chips:
GLUE Chip - MMU Memory Mngmnt Unit
DMA Controller - Graphics Processing Unit
16/32 bit Motorola 68000 processor at 8MHz
eight 32 bit data registers
eight 32 bit address registers
16 bit data bus/24 bit address bus
7 levels of interrupts/56 instructions
14 addressing modes/5 data types

DATA STORAGE
High speed hard disk interface
Direct memory access 1.33 Mbytes per second
CD (Compact Disc) interface
Built in cartridge access
Dedicated floppy disk controller

DISK DRIVE
500K (unformatted) s/sided 3½" floppy d/drive - 349K (formatted) storage capacity

SOUND AND MUSIC
Sound Generator
Frequency control from 30Hz to above audible
3 voices (channels) in wave shaping sound in addition to a noise generator
Separate frequency and volume controls
Dynamic envelope controls
ADSR (Attack, Decay, Sustain, Release)
Noise generator
MIDI interface for external music synthesizers

KEYBOARD
Separate keyboard microprocessor
Standard QWERTY typewriter styling
Ergonomic angle and height
95 keys including 10 function keys
Numeric keypad — 18 keys including ENTER
One touch cursor control keypad

MONITOR
12" screen - high res monochrome monitor
640x400 monochrome resolution

VIDEO PORTS
Display - Low Resolution - 40 columns
Med/High Res - 40/80 plus cols
Medium res RGB (Red/Green/Blue) output
High resolution monochrome (Black & White)

COMMUNICATIONS
Bidirectional centronics parallel interface for printers, or modems capable of input/output
RS232C serial modem/printer interface
VT52 Terminal Emulation Software
Maximum Baud Rate up to 19,200
High speed hard disk interface
Floppy disk controller (Western Digital)
2 joystick ports (one for 2 button mouse)
MIDI interface for external music synthesizers

GEM WIMP ENVIRONMENT
WIMP - Window Icon Mouse Pop-down menus
Two button mouse controller
Icons/Pull down menus/Windows
GEM VDI - Virtual Device Interface
GEM AES - Application Environment Services
GEM BBT - Bit Block Transfer
Real time clock & calender

SOFTWARE
GEM environment
with user friendly Macintosh style operation
'TOS' - Tramiel Operating System
Atari's own system based on CP/M 68K with hierarchical directory & file structure plus a host of MS DOS & UNIX command structures
'BOS' - Business Operating System
to run any standard BOS business programs
GEM desktop
with GEM PAINT graphics mgmt system and GEM WRITE word processor
Personal BASIC and DR Logo
originally written by Digital Research (DR) Very much like those on other machines except for the extensive use of pull down menus, mouse control and windows

VARIOUS
Dimensions: 470mmx240mmx60mm
Replaceable external power supplies
Expansion: 3½" floppy disk drives 500K/1,000K (two drives can be connected)
3½" 15Mb hard disk
CD (compact laser disc)
Dot matrix & d/wheel prntrs (black)
Thermal dot matrix (colour)
RGB & monochrome monitors

LANGUAGES
BASIC & LOGO supplied
Many others will soon be available, including: Assembler, BCPL, C, Cobol, Compiled Basic Lisp, Modular-2 and Pascal

Note: Some of the above specifications are pre-release and may therefore be subject to change

MACINTOSH v F1e v 520ST

"Imagine a Fat Mac - the 512K Apple Macintosh - but with a bigger screen, a far bigger keyboard with numeric keypad, cursor and function keys, and colour. That gives you some idea of what the Atari 520ST is like, except for two important things. First the Atari seems faster. Second the Atari system is about one third of the price."
June 1985 - Jack Schofield - **PRACTICAL COMPUTING**

FEATURES OF BASIC SYSTEM	APPLE MACINTOSH	APRICOT F1e	ATARI 520ST
Price Includes B/W Monitor	YES	NO - extra £200	YES
Keyboard size mm (LxDxH)	330x147x50	450x167x28	470x240x60
Keyboard size ins (LxDxH)	13x5¾x2	17½x6½x1	18½x9½x2½
3½" D/Drive (Unformatted)	500K	500K	500K
3½" D/Drive (Formatted)	399K	315K	349K
WIMP (Window, Icon, Mouse ...)	Apple	ACT - Activity	GEM
Real-time Clock	YES	YES	YES
Polyphonic Sound Generator	YES	NO	YES
RS232 Serial Port	YES	YES	YES
Centronics Parallel Printer Port	NO	YES	YES
Dedicated Floppy Disk Controller	NO	YES	YES
Hard Disk DMA Interface	NO	YES	YES
Full stroke keyboard	YES	YES	YES
Number of keys on keyboard	59	92	95
Numeric Keypad	NO	YES (16 Keys)	YES (18 keys)
Cursor Control Keypad	NO	YES	YES
Function keys	NO	10	10
16-bit processor	68000	Intel 8086	68000
Processor running speed	8MHz	4.77MHz	8MHz
RAM size	512K	256K	512K
Number of graphics modes	1	4	3
Number of colours	Monochrome	16	512
Max Screen Resolution (pixels)	512 x 342	640 x 256	640 x 400
Mouse included	Single Button	NO - extra £95	Two Button
Replaceable External Power Pack	NO	NO	YES
Cartridge Socket	NO	NO	YES
Joystick Ports	NO	NO	YES (two)
MIDI Synthesiser Interface	NO	NO	YES
Monitor Size	9"	9" - extra £200	12"
RGB Video Output	NO	YES	YES

System Cost with: Mouse - Monochrome Monitor - 512K RAM - 500K Disk Drive

	APPLE MACINTOSH	APRICOT F1e	ATARI 520ST
Price of basic system (exc VAT)	£2595+VAT	£595+VAT	£652+VAT
+ Mouse	Included	£95+VAT	Included
+ Monochrome Monitor	Included	£200+VAT	Included
+ Expansion to 512K RAM	Included	£295+VAT	Included
Price of complete system (exc VAT)	£2595+VAT	£1185+VAT	£652+VAT
PRICE rounded down including VAT	£2,984	£1,362	£749

PRESS COMMENT

"Atari's new corporate image as an aggressive low cost computer maker is likely to mirror that of Commodore where Mr. Tramiel established the maxim that 'Business is war'."
August 21st 1984 **FINANCIAL TIMES**

"This is the only personal computer I know of that comes with a Midi interface as standard."
Peter Bright March 1985 **PERSONAL COMPUTER WORLD**

"The (GEM) version running on the Atari 68000 machines will have the additional advantage of leaving the PC version standing." April 6th 1985 **PERSONAL COMPUTER NEWS**

"It would seem that GEM offers the ideal operating system."
March 7th 1985 **POPULAR COMPUTING WEEKLY**

"I found it (GEM) extremely easy to use and was very impressed with the way in which it disguises the unfriendly hardware and operating systems lurking under the surface."
Peter Bright Feb 1985 **PERSONAL COMPUTER WORLD**

"The electronics in the machine are a work of art ... The heart of the 520ST is a Motorola 68000, one of the most powerful 16-bit processors around and in many respects it is close to being a 32-bit chip ... when the machine appears in the shops, I'll be at the front end of the queue to buy one."
Peter Bright June 1985 **PERSONAL COMPUTER WORLD**

"This machine is significantly more powerful than an IBM PC ... If it's possible to design a sure-fire winning machine, this is it." May 11th 1985 **PERSONAL COMPUTER NEWS**

"... the use of GEM makes the new range of Atari computers so similar to the Macintosh (with the added attraction of colour), that they are already being called 'Jackintoshes'."
May 2nd 1985 **COMPUTING**

"The new Atari ST computers truly represent to the consumer what Jack Tramiel is saying — easy-to-use computing power without the price." March 1985 **ANALOG COMPUTING**

"It (the ST) uses the most modern technology that is affordable, in a package that gives a professional impression."
May 23rd 1985 **POPULAR COMPUTING WEEKLY**

"The Atari ST is one of the most elegant designs I have seen ... Atari has used an original and elegant method of memory management which should make the ST faster than any other PC on the market - in any price bracket ... The 64K dollar question is would I go out and spend money for one? To which the only answer is 'Try and stop me!'"
John Lambert July 1985 **ELECTRONICS & COMPUTING**

"The 520ST is technically excellent ... The 520ST hardware is the new standard by which others will be judged."
July 1985 **YOUR COMPUTER**

ATARI SILICA SHOP ATARI

WE ARE THE UK's No1 ATARI SPECIALISTS

At Silica we have been successfully dedicated to Atari ever since their products first appeared on the UK market. We can attribute our success largely to the Atari specialisation which we practice and to the user back-up we provide. Rest assured that when you buy a piece of Atari hardware at Silica you will be fully supported. Our mailings giving news of software releases and developments will keep you up to date with the Atari market and our technical support team and sales staff are at the end of the telephone line to deal with your problems and supply your every need. With our specialist bias, we aim to keep stocks of all the available Atari hardware, software, peripherals and accessories. We also stock a wide range of Atari dedicated books and through us, the owners on our list can subscribe to several American Atari dedicated magazines. We can provide a full service to all Atari owners and are now firmly established as the UK's NUMBER ONE Atari specialists. Here are just some of the things we can offer to our customers.

- **FREE POST & PACKING ON MAIL ORDERS**
- **FREE NEXT DAY SECURICOR DELIVERY**
- **INFORMATION MAILING SERVICE**
- **TECHNICAL SUPPORT TEAM**
- **HIGHLY COMPETITIVE PRICES**
- **AFTER SALES SUPPORT SERVICE**
- **REPAIR SERVICE ON ATARI PRODUCTS**

If you would like to be registered on our mailing list as an Atari computer owner, or as a person interested in buying an Atari machine, let us know. We will be pleased to keep you up to date with new Atari developments free of charge. So, return the coupon today and begin experiencing a specialist Atari service that is second to none.

SILICA HOTLINE 01-309 1111

SILICA SHOP LTD, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

SEND FOR FREE ATARI ST LITERATURE

To: Silica Shop Ltd, Dept ATUSR 0885, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

PLEASE SEND ME FREE LITERATURE

ON THE NEW ATARI 520ST COMPUTER

Mr/Mrs/Ms: .... Initials: .... Surname: ....

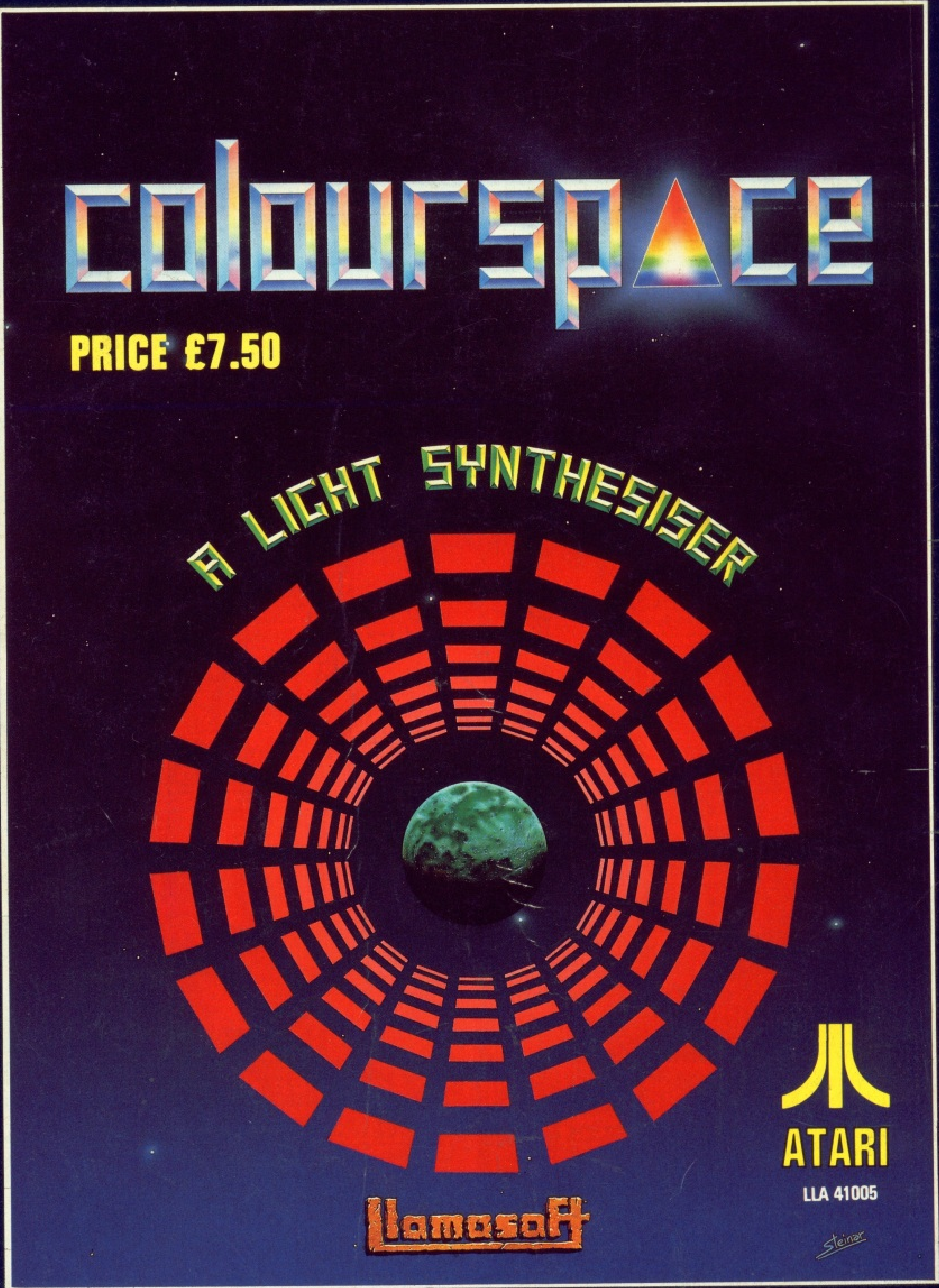
Address: ....

....

....

.... Postcode: ....

Do you already own a computer
If so, which one do you own? ....



colourspace
PRICE £7.50
A LIGHT SYNTHESISER
ATARI
LLA 41005
Llamasoft